UNIVERSIDAD DE CÓRDOBA

Departamento de Informática y Análisis Numérico

Utilizando métodos de descomposición, algoritmos kernel y técnicas de remuestreo para clasificación ordinal

Doctorado internacional
Programa de doctorado: Ingeniería y Tecnología

María Pérez Ortiz

Directores
César Hervás Martínez

Pedro Antonio Gutiérrez Peña

Departamento de Informática y Análisis Numérico
Córdoba, enero de 2014

TITULO: *Utilización de métodos de descomposición, algoritmos kernel y técnicas de remuestreo para clasificación ordinal*

AUTOR: *María Pérez Ortiz*

Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

www.uco.es/publicaciones
publicaciones@uco.es

UNIVERSITY OF CÓRDOBA

Department of Computer Science and Numerical Analysis

Exploiting decomposition methods, kernel algorithms and over-sampling techniques for ordinal regression

International Doctorate
Program: Engineering and Technology

María Pérez Ortiz

Supervisors
César Hervás Martínez

Pedro Antonio Gutiérrez Peña

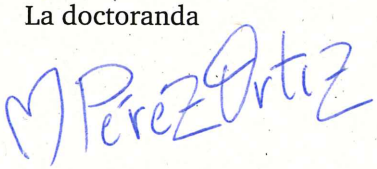
Department of Computer Science and Numerical Analysis
Córdoba, January 2014

La memoria titulada "*Exploiting decomposition methods, kernel algorithms and over-sampling techniques for ordinal regression*", que presenta D.ª María Pérez-Ortiz para optar al grado de Doctora, ha sido realizada dentro del programa de doctorado "Ingeniería y Tecnología" del Departamento de Informática y Análisis Numérico de la Universidad de Córdoba bajo la dirección del Doctor D. César Hervás Martínez y del Doctor D. Pedro Antonio Gutiérrez Peña.

La doctoranda D.ª María Pérez-Ortiz y los directores de la tesis D. César Hervás Martínez y D. Pedro Antonio Gutiérrez Peña garantizamos, al firmar esta Tesis Doctoral, que el trabajo ha sido realizado por la doctoranda, bajo la dirección de los directores de la Tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

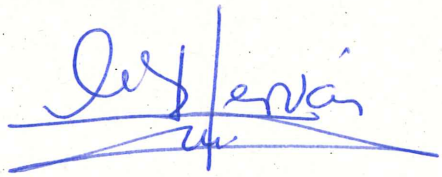
Córdoba, diciembre de 2014

La doctoranda



Fdo: María Pérez-Ortiz

Los directores



Fdo: César Hervás Martínez

Fdo: Pedro Antonio Gutiérrez Peña

TÍTULO DE LA TESIS: Exploiting decomposition methods, kernel algorithms and over-sampling techniques for ordinal regression

DOCTORANDA: María Pérez Ortiz

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

La presente tesis doctoral aborda la resolución de problemas de clasificación ordinal (también llamados de regresión ordinal, por rango o por ordenación), siendo éste un problema de aprendizaje supervisado de predicción de categorías en una escala ordinal. De forma similar a lo que sucede en los problemas de clasificación nominal, las muestras son etiquetadas mediante un conjunto de categorías, pero en el caso de la clasificación ordinal, estas categorías están ordenadas. De forma más específica, los puntos fundamentales abordados en la tesis doctoral son los siguientes:

- Una revisión y comparación de técnicas de clasificación ordinal que dan idea del estado actual de las mismas.
- La exploración en profundidad de técnicas de descomposición del problema multiclase ordinal en problemas biclase.
- El desarrollo de técnicas específicas de aprendizaje de funciones de *kernel* para problemas de clasificación ordinal.
- La *kernelización* de métodos lineales estándar de clasificación ordinal.
- El desarrollo de nuevos algoritmos de remuestreo de patrones para bases de datos no balanceadas, generando los patrones de acuerdo al espacio empírico de características que define la función de *kernel* a utilizar.
- La propuesta de nuevos métodos de resolución del problema de desbalanceo de las clases, cuando éstas están en escala ordinal.
- La aplicación de estas técnicas y metodologías a la resolución de problemas complejos e interesantes como la determinación del mejor órgano a trasplantar a pacientes en cola de espera, teniendo en cuenta el tiempo estimado (medidos en clases ordinales) de supervivencia al trasplante; la clasificación ordinal de países en función de su adaptación a un desarrollo sostenible; y otros problemas adicionales.

El trabajo de la doctoranda en la temática de esta tesis doctoral comenzó a mediados del 2011, con la realización de su proyecto fin de carrera, y posteriormente su trabajo fin de master. La evolución de la tesis desde entonces ha sido muy rápida y fructífera. En esos primeros año, la doctoranda ya mandó un primer trabajo de clasificación ordinal al congreso *Intelligent Systems Design and Applications* (ISDA). A partir del año 2012 y hasta la fecha, el número de trabajos publicados en revistas de alto índice de impacto, así como en congresos internacionales de primer nivel ha sido constante. En el año 2012, la doctoranda presentó sus trabajos e ideas en varios congresos y mandó varios trabajos a revistas.

Durante el curso académico 2013/2014, la tesis es inscrita en el programa de doctorado y la productividad, y con ella, el número de trabajos ha tenido un aumento exponencial. Prueba de ello es la extensa bibliografía que se muestra a continuación.

Reseñar por último que es una tesis extensa con muchas ideas novedosas y que abre varios caminos a la comunidad investigadora para poder seguir transitando por ellos. También cabe destacar que algunos de los temas abordados en la tesis (trabajar en espacios de características, utilizando estructuras de tipo *manifold* o la utilización de grafos para analizar los patrones vecinos siguiendo el orden de las clases) requieren un conocimiento profundo de conceptos pertenecientes a áreas como la Estadística, el Análisis funcional y la Geometría.

Conocimiento que pocos doctorandos en Ciencias de la Computación suelen tener, ya que a éstos se les supone además una capacidad sobresaliente de programación y de gestión de procesos, áreas en las que la doctoranda también ha destacado sobradamente.

Durante el desarrollo de la tesis doctoral, la doctoranda ha asistido a eventos científicos (congresos) de relevancia internacional y nacional y ha publicado artículos en revistas científicas, los cuáles se relacionan a continuación.

Artículos en revistas:

[J1] M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás Martínez. Projection-Based Ensemble Learning for Ordinal Regression. IEEE Transactions on Cybernetics, 44(5):681–694, 2014, Impact Factor (2013): 3.781 (Q1).

[J2] M. Pérez-Ortiz, M. Cruz-Ramírez, M. D. Ayllón-Terán, N. Heaton, R. Ciria and C. Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. Applied Soft Computing, 14:88–98, 2014, Impact Factor (2013): 2.679 (Q1).

[J3] M. Pérez-Ortiz, M. de la Paz-Marín, P.A. Gutiérrez and C. Hervás-Martínez. Classification of EU countries' progress towards sustainable development based on ordinal regression techniques. Knowledge-Based Systems, 66:178–189, 2014, Impact Factor (2013): 3.058 (Q1).

[J4] M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelising the Proportional Odds Model through Kernel Learning techniques. Neurocomputing, In press, 2014, Impact Factor (2013): 2.005 (Q1).

[J5] M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez and X. Yao. Graph-Based Approaches for Over-sampling in the context of Ordinal Regression. IEEE Transactions on Knowledge and Data Engineering (TKDE), In press, 2015, Impact Factor (2013): 1.815 (Q1).

Artículos en revistas (bajo revisión):

[J6] P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro and C. Hervás-Martínez. Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering (Under Review), 2014, Impact Factor (2013): 1.815 (Q2).

[J7] M. Pérez-Ortiz, M. Fernández-Delgado, E. Cernadas, R. Domínguez-Pétit, P.A. Gutiérrez and C. Hervás-Martínez. On the use of nominal and ordinal classifiers for the discrimination of states of development in fish oocytes. Neural Processing Letters (Under Review), 2014, Impact Factor (2013): 1.237 (Q2).

[J8] M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. A study on multi-scale kernel optimisation via centred kernel-target alignment. Neural Processing Letters (Under Review), 2014, Impact Factor (2013): 1.237 (Q2).

[J9] M. Pérez-Ortiz, P.A. Gutiérrez, P. Tino and C. Hervás-Martínez. Over-sampling the minority class in the feature space. Submitted to IEEE Transactions on Neural Networks and Learning Systems (Under Review), 2014, Impact Factor (2013): 4.370 (Q1).

Congresos:

[C1] M. Pérez-Ortiz, P.A. Gutiérrez, C. García-Alonso, L. Salvador-Carulla, J.A. Salinas-Pérez y C. Hervás-Martínez. Ordinal classification of depression spatial hot-spots of prevalence. In Intelligent Systems Design and Applications (ISDA), pages 1170–1175, 2011.

[C2] P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez. An experimental study of different ordinal regression methods and measures. In 7th International Conference on Hybrid Artificial Intelligence Systems, pages 296–307, 2012.

[C3] M. Pérez-Ortiz, L. García-Hernández, L. Salas-Morera, C.Hervás-Martínez, and A. Arauzo-Azofra. An ordinal regression approach for the unequal area facility layout problem. In Soft Computing Models in Industrial and Environmental Applications (SOCO), pages 13–21, 2012.

[C4] M. Pérez-Ortiz, A. Arauzo-Azofra, C. Hervás-Martínez, L. García-Hernández y L. Salas-Morera. A system learning user preferences for multiobjective optimization of facility layouts. In Soft Computing Models in Industrial and Environmental Applications (SOCO), pages 43–52, 2012.

[C5] M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, J. Briceño y M. de la Mata. An ensemble approach for ordinal threshold models applied to liver transplantation. In International Joint Conference on Neural Networks (IJCNN), pages 2795–2802, 2012.

[C6] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, and C. Hervás-Martínez. Estudio comparativo de distintos m´todos de umbral en regresi´n ordinal. In IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB), pages 872–881, 2013.

[C7] M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. Multiscale support vector machine optimization by kernel-target alignment. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 391–396, 2013.

[C8] M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Synthetic over-sampling in the empirical feature space. In European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pages 385–390, 2013.

[C9] M. Pérez-Ortiz, R. Colmenarejo, J.C. Fernández-Caballero y C. Hervás-Martínez. Can machine learning techniques help to improve the common fisheries policy?. In International Work Conference on Artificial Neural Networks (IWANN), Lecture Notes in Computer Science Volume 7903, pages 278–286, 2013.

[C10] M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelizing the proportional odds model through the empirical kernel mapping. In International Work Conference on Artificial Neural Networks (IWANN), Lectures Notes in Computer Science Volume 7902, pages 270–280, 2013.

[C11] M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Borderline kernel based oversampling. In International Conference on Hybrid Artificial Intelligence Systems (HAIS), Lecture Notes in Computer Science Volume 8073, pages 472–481, 2013.

[C12] M. Pérez-Ortiz, P.A. Gutiérrez y C. Hervás-Martínez. Log-gamma distribution optimisation via maximum likelihood for ordered probability estimates. In International Conference on Hybrid Artificial Intelligence Systems (HAIS), Lecture Notes in Computer Science Volume 8480, pages 454–465, 2014.

[C13] M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Learning kernel label decompositions for ordinal classification problems. In International Conference on Neural Computation Theory and Applications, pages 218–225, 2014.

[C14] M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Incorporating privileged information to improve manifold ordinal regression. In International Conference on Neural Computation Theory and Applications, pages 187–194, 2014.

Por todo lo cual, consideramos que la tesis doctoral presentada reúne las condiciones de originalidad y rigor científico necesarias y cuenta con los avales necesarios para su exposición y defensa.

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 15 de diciembre de 2014

Firma del/de los director/es

Fdo.: César Hervás Martínez

Fdo.: Pedro Antonio Gutiérrez Peña

Esta Tesis Doctoral ha sido financiada en parte con cargo a los Proyectos **TIN2011-22794** de la Comisión Interministerial de Ciencia y Tecnología (CICYT), el Proyecto de Excelencia **P11-TIC-7508** de la Junta de Andalucía y con fondos FEDER.

This work has been partially subsidized by the **TIN2011-22794** project of the Spanish Ministerial Commission of Science and Technology (CICYT), FEDER funds and the **P11-TIC-7508** project of the "Junta de Andalucía" (Spain).

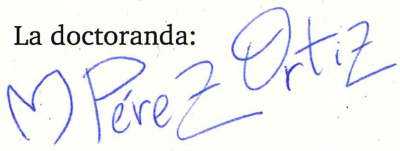
Mención de Doctorado Internacional

Esta tesis cumple los criterios para la obtención de la mención «Doctorado Internacional» concedida por la Universidad de Córdoba. Para ello se presentan los siguientes requisitos:

1. Estancia predoctoral realizada en otros países europeos:
   - **School of Computer Science, University of Birmingham, Birmingham, Reino Unido.** 3 meses de mayo a agosto de 2012. Tutor de la estancia: **Dr. Peter Tiňo**, *Full professor* de Escuela de Ciencias de la Computación en la Universidad de Birmingham (Reino Unido).
2. Esta tesis está avalada por los siguientes informes de idoneidad realizados por doctores de otros centros de investigación europeos:
   - **Dr. Huanhuan Chen**. *Professor* en la Escuela de Ciencias de la Computación de la Universidad de Ciencia y Tecnología de China (China).
   - **Dr. Felipe Martins Müller**. *Visiting Professor* en la Escuela de Ciencias de la Computación e Informática de la Universidad de De Montfort (Reino Unido).
3. La defensa de tesis y el texto se han realizado totalmente en inglés.
4. Entre los miembros del tribunal se encuentra un doctor procedente de un centro de educación superior europeo, tratándose del Dr. **David Elizondo**, *Professor* de la Escuela de Ciencias de la Computación e Informática de la Universidad de De Montfort (Reino Unido).

Córdoba, Diciembre 2014

La doctoranda:



Fdo.: María Pérez Ortiz

The "adas"[1]*, prefer the structure "switch... case" over the structure "if... else", given its infinite possibilities. Programming the world for the "adas" implies creating the conditions for each one to be whatever one wants to be, even when this corresponds to something not referenced yet with a word in the world, or when it can not be imagined clearly; it implies creating the conditions that even allow us to change our mind.*

Las "adas"[2]*, frente a la estructura "if... else...", prefieren la estructura "switch... case..." llevada a una infinidad de opciones. Programar el mundo para las "adas" implica crear las condiciones para que cada cual pueda ser lo que uno mismo quiera ser, incluso cuando lo que se quiera corresponda a lo que aún no está referenciado con una palabra en el mundo, o todavía no se imagina con claridad; implica crear las condiciones que incluso nos permitan cambiar de opinión.*

Remedios Zafra. "(h)adas, mujeres que crean, programan, prosumen, teclean". 2013

[1] "Adas" making reference to the generations of women who have followed Ada Byron in its desire of breaking the mold, of creating and programming. Note that hada in Spanish (pronounced the same as ada) means fairy.

[2] "Adas" hace referencia a las generaciones de mujeres que han seguido los pasos de Ada Byron en su deseo de romper moldes, de crear y programar.

“What you do makes a difference, and you have to decide what kind of difference you want to make.”

Jane Goodall

To all the billions of animal souls that expire every year in great pain for satisfying our greed and our fleeting pleasure. To all the children that do not get to know the world because of this, because we look the other way and do not connect the dots. I dream of the day when we finally stop torturing all of you.

To all the women who fought incandescently with their life for the rest of us to be free. Actually, to all the people who fought and is fighting at this moment. You are my inspiration, do not let this bitter world discourage you and steal your dreams.

“Lo que haces marca una diferencia, pero tienes que decidir qué tipo de diferencia quieres marcar.”

Jane Goodall

A las billones de almas animales que expiran cada día envueltas en dolor para satisfacer nuestra codicia y nuestro placer pasajero. A todos los niños que no llegan a conocer el mundo a causa de esto, a causa de que miremos hacia otro lado y nos empeñemos en no conectar los puntos. Sueño con el día en que por fin dejemos de torturaros a todos.

A todas las mujeres que han luchado incandescentemente con su vida para que el resto de nosotros seamos libres. En realidad, a todos los que han luchado y siguen luchando. Sois mi inspiración, no dejéis que este mundo amargo os desaliente y os robe los sueños.

A César y Pedro, por sus valiosas enseñanzas
y las muchas oportunidades que me han regalado.

A mis padres porque las semillas que plantaron
en mi con su educación nunca dejarán de germinar y crecer.

The story of the little kernel who wanted to surf the equations of life

Once upon a time, there was a little Gaussian kernel with a huge heart. He loved Mexican hats and spending time optimising itself. The little kernel had a big dream, to help everyone in their machine learning tasks. He wanted to help meteorologists to predict possible snowfalls, so children could know when they could make a snowman. He wanted to help people to detect outliers, such as true love or real friendship. He wanted to help the human being to understand what dolphins or birds say, but mostly, he wanted to help them to distinguish trivial things from really important ones, such as visiting Mother Earth or dancing under the light of the stars.

His father, a linear kernel, thought it was pointless. His mother, a polynomial kernel, thought it was too complicated to please everyone. “I know that, mum” - said the little kernel - “I know there is a huge search space out there with hundreds of local optima but... wouldn’t it be wonderful to embark on such an adventure and become an universal approximator?”.

Finally, both dad and mum gave up and so, the little kernel made his way to the kernel school. There he spent the best years of his life. He loved to play with the support vectors, and there he knew a lovely and astonishing accurate projection, which was later to become his partner in life. He learnt everything about optimisation algorithms, and so, with some time, he became the first of his class, being always the first in finding the best solution to every problem. The little kernel finally graduated a few years later and made a great scientist of himself. Now, he lives happily searching for new machine learning problems to solve.

Index

Figure index

We are all apprentices in a craft where no one ever becomes a master.

Ernest Hemingway

1

Introduction, motivation and objectives

As science and technology have been achieving a vast amount of important and useful discoveries, our lifestyle has also changed. All this progress has incredibly improved our standard of living, allowing an increased industrial production, relieving the human being from some mechanical and tedious works, and enabling communication from one point to the other of the world, among others.

The classification paradigm has always been one of the more complex but useful tasks for humans. It allows us to impose some order on reality and understand it (at least slightly better), either for classifying animals, different kinds of vegetation, diseases, etc. But most of these problems are solved at a sensory level or intuitively, without an explicit method or algorithm (up to our understanding). Nonetheless, nowadays the large amount of valuable and available data has become intractable. Because of that, the development and application of new automatic processing techniques has become necessary, in order to extract proper information and knowledge from this enormous quantity of data. As a result of this imperious necessity of developing specialised processing techniques, some new research branches have emerged under the general notion of data science (as pattern recognition, machine learning and data mining).

The goal of using these data is to build systems adaptable to their environments and that can learn from their experience, an issue which has attracted researchers from many different fields, including computer science, engineering, physics, mathematics, neuros-

cience and cognitive science. Out of this research, a wide variety of learning techniques have appeared with the enough potential to transform many scientific and industrial fields.

It could be said that "*if data had mass, the earth would be a black hole*" [81], because, all around the world, computers capture and store terabytes of data every day. In this sense, banks are building up pictures of how people spend their money, hospitals are recording information about patients (e.g. diseases, corresponding treatments used, responses to medication, etc.), and engine monitoring systems in cars are recording information about the engine in order to detect when it might fail. The challenge then is how to exploit this data in order to do something constructive that leads to a improvement in our society. The main question is, if bank's computers could learn about spending patterns, can they help to detect credit card fraud quickly? If hospitals share their data and make use of it, then, could treatments that do not work as well as expected be identified quickly? Could an intelligent car give you early warnings of problems so that you do not end up stranded in the worst part of town? These are some of the questions that data science methods (and more specifically machine learning) can be used for. The key concept in this case is learning from data, which in terms of human behaviour might be seen as learning from experience. The more important parts of human learning in this case are memory and ability to adapt and generalise. One of the most interesting features of machine learning is that it lies on the boundary of several different academic disciplines, principally computer science, statistics, mathematics and engineering, although it also makes use of concepts of nature and the phisiology of the human brain.

1.1. Data science and machine learning

Generally, the term data science refers to the extraction of knowledge from data. This involves a wide range of techniques and theories drawn from many research fields within mathematics, statistics and information technology, including statistical learning, data engineering, pattern recognition, uncertainty modelling, probability models, high performance computing, signal processing, and machine learning, among others. Precisely, the growth and development of this last research area (i.e. machine learning) has made data science more relevant, increasing the necessity of data scientists and the development of novel methods in the scientific community, given the great breadth and diversity of knowledge and applications of this area.

It is felt that the decision-making processes of a human being are somewhat related to the recognition of patterns [41] (it is usually not directly stated because at the moment the complex behavioural patterns of the human brain remain still unresolved). For instance, the next move in a chess game is decided upon the present pattern on the board, analysing potential and auspicious moves and anticipating what our opponent might do.

The goal of pattern recognition is to emulate these complicated mechanisms of decision making processes and automate them using computers (in order to simplify some intractable or tedious tasks). However, because of the complex nature of this emulation, most pattern recognition and machine learning research is actually focused on more realistic and practical problems, such as the ones represented in Figure 1.1.1: applications in biometry such as iris or fingerprint recognition, handwriting or plate identification, facial recognition, brain segmentation and others. However, there are other research projects in machine learning much more ambitious than simplifying some aspect of our lives. One example is the Human Brain Project, which aims to simulate the complete human brain on supercomputers to better understand how it works.

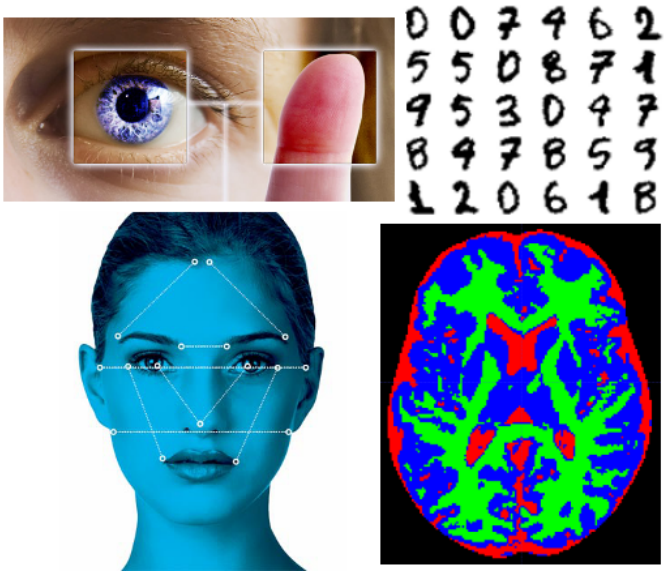


Figure 1.1.1: Examples of machine learning applications: biometry, handwriting recognition, facial recognition and subcortical segmentation.

Machine learning methods can be divided according to different criteria, as the following taxonomy shows[1] (and also Figure 1.1.2):

- Concerning the nature of the model: whether it is a full or partial probabilistic model or it fits a discriminant/regression function directly.
- Considering the type of reasoning applied: inductive or transductive depending on whether the model performs a reasoning from observed training cases to general rules or the other way around.
- According to the learning task itself: in this sense, we can distinguish between supervised, unsupervised or reinforcement learning.

[1]Note that there might be very different categorisations for the machine learning methods, but these are, to our opinion, the most general ones.

- Regarding the manner in which the training data are presented to the learner: batch learning, when all the data are given to the learner at the beginning of the process; or online learning, when the learner receives one example at a time (in this case the learner updates its current hypothesis in response to each new example).
- Taking into account the type of learning task to perform: e.g., classification or regression (although this division can be subdivided in more complex and different tasks).
- Depending on the classification model itself: generative or discriminative, depending on whether it defines the joint probability of the data and latent variables of interest, and therefore explicitly states how the observations are assumed to have been generate; or it simply focuses on discriminating one class from another.

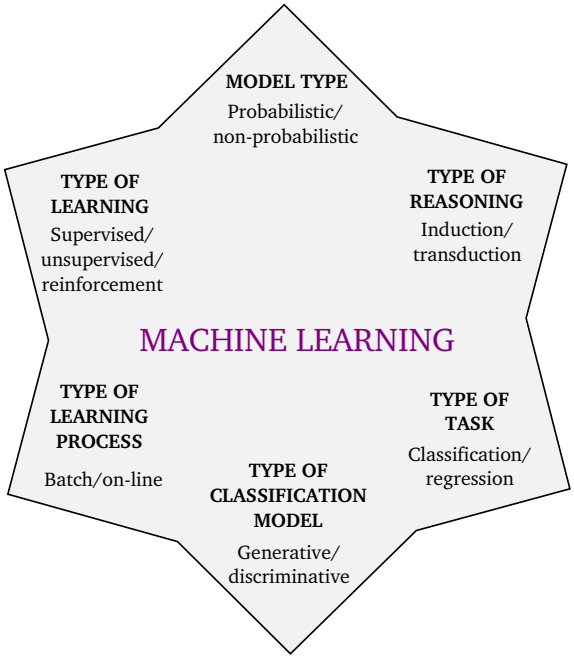


Figure 1.1.2: Different categorisation for the machine learning paradigm.

As stated, machine learning methods can be categorised and framed under different criteria. However, there are some categorisations that are of more interest to the potential reader of this thesis. The differentiation of the concepts supervised and unsupervised learning [11], is, for example, of special interest. Unsupervised learning [52, 60] tries to discover the structure of the clusters underlying the data from purely unlabelled data. This paradigm is generally useful to analyse whether there are differentiated groups or clusters present in the data. The curse and the blessing of this branch of pattern recognition is that there is no ground truth against which to compare the results, i.e. there is not a single wrong clustering structure because there are many different ways to cluster a set

of patterns. However, for supervised learning, each object in the data is preassigned to a class label before the learning process, and this information is given to the learning algorithm [49]. Then, the main task is to design an algorithm able to learn a classification rule that ideally produces the same labelling for the provided data. Most often, the labelling process can not be described in an algorithmic form. Thus, we supply the machine with learning skills and present the labelled data to it. The classification knowledge learned by the machine in this process might be obscure, but the recognition accuracy of the classifier will be the judge of its adequacy.

Other important categorisation relies on the nature of the labelling space (only applicable for supervised learning). Under this premise, one can divide the methods in classification or regression techniques. Subsequent divisions of these terms can be also done: multi-label classification, multi-instance classification, multiple-output regression, etc. However, there is one learning paradigm, known both as ordinal classification or ordinal regression, which lies between the classification and regression paradigms. In fact, the concept of ordinal classification, which will be discussed and exposed in the following sections, is the cornerstone of this thesis.

The following subsections briefly introduce some of the main topics of the thesis, separating between methodology (different paradigms of classification and kernel techniques) and real-world applications.

1.1.1. Classification

Supervised learning is perhaps the most common learning problem considered in machine learning. For instance, suppose we want to predict house prices. Under this premise, we can collect data regarding house prices and information such as its size in square meters, the number of bedrooms, how old the house is, and others in order to provide the algorithm with this training data which is composed of the independent variables (size, number of bedrooms, etc) and the dependent variable (the price itself). The learning algorithm will be then responsible for the construction of an accurate learning model from this labelled data. In this case, since the output variable is continuous, rather than discrete, the learning problem is referred to as regression. However, when this labelling space, or output variable is discrete, the learning problem receives the name of classification.

Typically, for the purpose of classification, an unique dependent variable is considered. However, other supervised classification branches exist, such as multi-output, multi-label, multi-instance or one-class classification, which are slightly different learning paradigms. The general aim of the classification algorithm is to be able to separate the classes of the problem (as far as possible) using only the information provided by the training data. If the output variable has two possible values, the problem is referred to as binary

classification. On the other hand, if there are more than two classes, the problem is named as multiclass or multinomial classification. There are other categorisations of classification methods, and two of the main ones are the focus of the study of this thesis, i.e. ordinal classification and imbalanced classification.

Ordinal classification

The classification of patterns into naturally ordered labels is usually referred to as ordinal regression or ordinal classification. This learning paradigm, although still mostly unexplored, is spreading rapidly and receiving a lot of attention from the pattern recognition and machine learning communities, given its applicability to real world problems (e.g. economy, medicine, psychology and others). This paradigm can be said to lie between both classification and regression: as opposed to multinomial or multiclass classification, there exists some ordering between the categories in the labelling space $\mathcal{Y}$, and both standard classifiers and the common zero-one loss function do not capture and reflect this ordering; in contrast to regression, $\mathcal{Y}$ is a finite set and a non-metric space (i.e. distances between categories are unknown).

For an explanatory example of ordinal regression problems consider the case of articles/services rating via a Likert scale, where the categories of the problem (i.e. {strongly disagree, disagree, neither agree or disagree, agree, strongly agree}) corresponds to the level of agreement with a given statement (e.g. "do you agree that article/service x is helpful?"). In this case, the natural order among the classes can be appreciated, as well as the necessity of penalising differently the misclassification errors (it should not be considered equal misclassifying a "strongly disagree" pattern as one of the "strongly agree" class than misclassifying it as one of the "agree" class). The same problem that arises when classifying the level of severity of an illness using an ordinal scale.

An example of an ordinal synthetic dataset can be found in Figure 1.1.3, where, although the class labels are ordered, this order is only partially appreciated in the input space. The dataset is highly nonlinear, what motivates the use of nonlinear classifier.

More formally, the aim of ordinal regression classifiers is to learn a prediction rule $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^d$ corresponds to the input space and $\mathcal{Y}$ to the labelling space. Therefore, f will assign an input pattern $\mathbf{x}_i \in \mathcal{X}$ to one of the K discrete classes $\mathcal{C}_k, k \in \{1, \ldots, K\}, \mathcal{C}_k \in \mathcal{Y}$, where there exist a given ordering between the labels (i.e. $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$, $\prec$ denoting this order information). For the sake of understanding, denote to the independent and identically distributed (i.i.d.) training sample as $T = \{\mathbf{x}_i, y_i\}_{i=1}^N$, where N is the number of patterns.

Concerning ordinal problems, a common (although not totally correct) approach is to use nominal classifiers (obviating the ordinal information), regression methods (as-

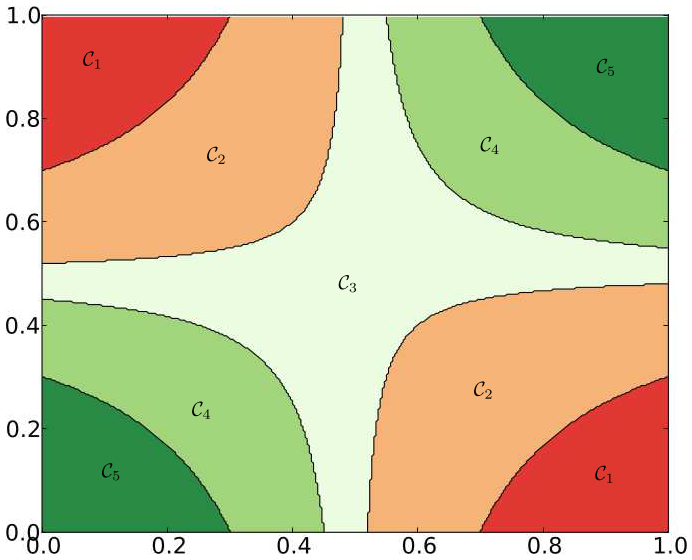


Figure 1.1.3: Two-dimensional ordinal toy dataset.

suming that the distances between different categories are known and equal) or a cost-sensitive approach (assuming, as we will see, the absolute cost when no further information is provided) [67]. Contrary to these approaches, the use of techniques specifically developed for ordinal regression has been proved to lead to better performance (in terms of the ordering of the classes) for multiple ordinal classification problems. Other approaches assess the problem by decomposing the original ordinal regression problem into a set of binary classification tasks [20, 39] (which, as we shall see, will also be one of the focus of the thesis), or by formulating the original problem as a larger augmented binary classification one [73]. However, the most popular way of tackling this kind of problem is using threshold models [101, 97, 84, 25]. These methods are based on the idea that, to model ordinal ranking problems from a regression perspective, one can assume that some underlying real-valued outcomes exist (also known as latent variable), but they are unobservable. Consequently, two different things are estimated:

- A function $f(\mathbf{x})$ that predicts the real-valued outcomes and intends to uncover the nature of the assumed underlying variable.
- A threshold vector $\mathbf{b} \in \mathbb{R}^{K-1}$ (where K is the number of classes in the problem) to represent the intervals in the range of $f(\mathbf{x})$, where $b_1 \leq b_2 \leq \ldots \leq b_{K-1}$ (possible different scales around different ranks).

To maintain and exploit the order information, most thresholds methods include a restriction on the projection to find. This is, these methods try to find the projection which provides the greater separation of the data but also maintains the classes ordered according to their rank (to minimise certain misclassification errors). To see this, analyse

Figure 1.1.4 where two projections can be seen ($\mathbf{w}$ and $\mathbf{w}'$). In this case, $\mathbf{w}'$ should be preferred over $\mathbf{w}$ because it maintains the data ordered according to their ranking ($\mathcal{C}_1 \prec \mathcal{C}_2 \prec \mathcal{C}_3 \prec \mathcal{C}_4$).

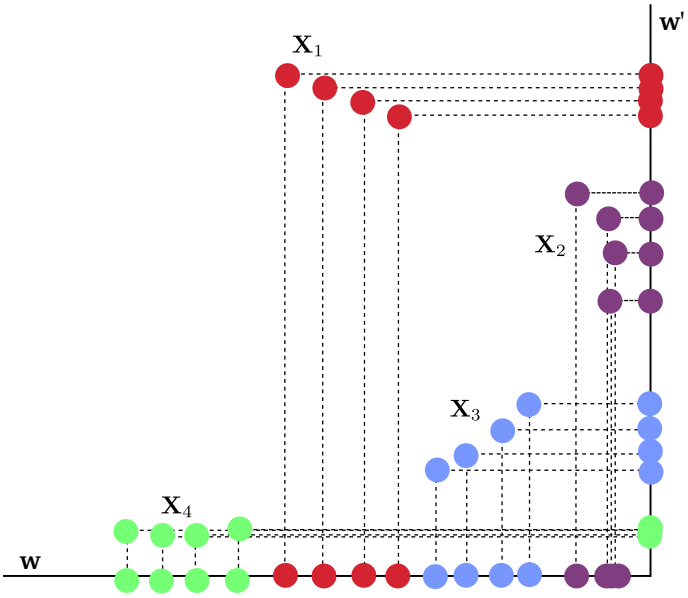


Figure 1.1.4: Different projections for an ordinal example.

Imbalanced classification

Machine learning techniques are usually based on the assumption that the target classes of the problem share similar prior probabilities. However, this is often not the case in many real-world applications in areas such as medical diagnosis, information retrieval, fraud detection, fault monitoring, etc. The classification paradigm when one or several classes have a much lower prior probability is known as imbalanced classification [51, 61] and it poses a real challenge for machine learning researchers. Because of that, imbalanced classification is currently receiving a lot of attention from the pattern recognition and machine learning communities [23, 102, 87, 18, 50, 10, 9]. Often, the minority class happens to be more important than the majority one, but it may also be much more difficult to model and identify complex underlying behaviour patterns due to the low number of available minority samples. Since most traditional learning systems have been designed to work on balanced data, they will usually be focused on improving overall performance and be biased towards the majority class, consequently harming the minority one [43]. Although, from a formal perspective, an imbalanced dataset is any set of labelled data exhibiting an unequal distribution between classes, it has been shown that this is not the only factor involved in hindering the learning in this context [51, 61]. The nature of the class imbalance problem can be also attributed to other factors, such as the complexity

of the data (existence of noisy and non-representative samples or class overlapping) or the size of the training set (high-dimensional data or small sample size). The approaches developed over the years for tackling the class imbalance problem can be categorised in two groups:

- First, the data approach, based on sampling methods, including over-sampling minority classes (groups of interesting rare examples), or under-sampling majority classes (groups with large example sizes), the combination of both being also very popular.
- Secondly, the algorithmic approach, which forces the classifier to pay more attention to the minority class, e.g. by a cost-sensitive approach [103].

Concerning previous studies, the cost-sensitive learning setting has been proved to lead to over-fitting [43] in some cases, thus data approaches are usually preferred. In the same vein, some studies suggest that over-sampling is more useful and powerful than under-sampling for highly imbalanced and complex datasets [61, 7], given the potential loss of meaningful information of under-sampling techniques.

Concerning over-sampling, as said, the first idea is to perform a random replication of the minority data, but this often leads to over-fitting [43]. Therefore, another common approach is to generate new synthetic patterns according to the minority class distribution. One of the most well-known methodologies in this group is the synthetic minority over-sampling technique (SMOTE) [23] based on generating new instances by convex combination of one point and one of its k-nearest neighbours (both belonging to the minority class). However, the classes can not be assumed to be convex in general, and hence SMOTE does not avoid new synthetic patterns to fall inside majority regions. Therefore, more careful techniques have been developed to prevent this issue (prevent, but not solve). Adaptive synthetic [87, 18, 50] and cluster-based sampling methodologies [9, 10] are examples of more flexible and powerful techniques, based on extracting knowledge directly from the data to analyse which patterns and regions of the space are more suitable for over-sampling. In this thesis, this will be referred to as preferential over-sampling.

At the same time, kernel methods have been spreading rapidly and gaining more acceptance from machine learning researchers due to their good generalisation ability and their determinism, being one of the most widely used the Support Vector Machine paradigm (SVM) [12, 28]. However, for the specific SVM technique, imbalanced data pose a serious challenge, due to the formulation of the soft-margin maximisation paradigm which is focused on improving overall performance. Thus, the combination of kernel methods with methods for tackling class imbalance is widely spread in the literature [102, 118].

It is clear that the problem of imbalanced classification also arises when tackling ordinal regression. For an explanatory example, consider the case of financial trading where

an agent intends to predict not only whether to buy an asset, but also the amount of investment. The different situations could be categorised as {“no investment”, “little investment”, “big investment”, “huge investment”}. In this case, the natural order among the classes can be appreciated, as well as the necessity of penalising differently the misclassification errors. However, one can also note that there are some classes that are naturally much more probable than others (specially when the number of classes is high), and, therefore, the problem present an imbalanced nature. According to this example, it is reasonable to expect a lower number of “huge investment” situations than the number of “little investment” ones. Although standard over-sampling methods could also be applied to ordinal regression, the new synthetic samples are obtained without taking into account the ordering scale, and this can result in classifiers more prone to commit errors of several categories in the ordinal scale, motivating this a further study on this topic.

1.1.2. Kernel methods

It is well-known that the formulation of a nonlinear separating hyperplane is much complex than the formulation of its linear version (in this sense, Figure 1.1.5 shows a linearly separable dataset and a nonlinearly separable one). Furthermore, it is widely known that linear real-world data is not the norm, but the exception. Therefore, the problem of handling the nonlinearity of the data has been vastly debated in machine learning.

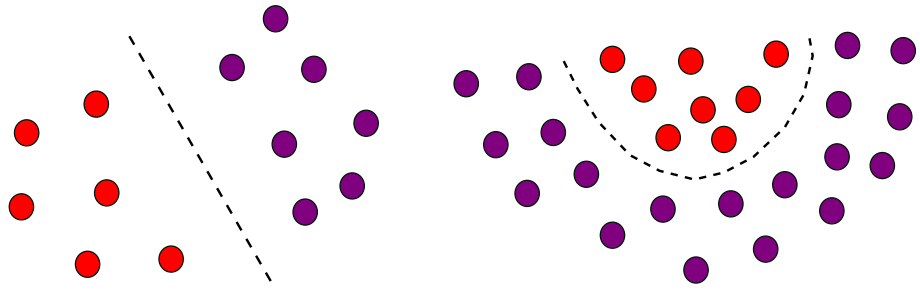


Figure 1.1.5: Linearly and nonlinearly separable datasets.

The idea is the construction of a nonlinear mapping function Φ which will map the data into a space where the classes are ideally linearly separable. In practice, this will translate to a nonlinear separating decision region in the original input space. Analyse Figure 1.1.6 for a one-dimensional explanatory example.

However, the choice of the mapping function Φ is also tricky, motivating this the use of kernel functions (i.e. instead of explicitly computing the function Φ, $\mathcal{H}$ can be efficiently obtained from a suitable kernel function). In this vein, it can be said that the crucial ingredient of kernel methodologies, such as kernel principal component analysis, kernel discriminant learning or support vector machines is undoubtedly the application of the

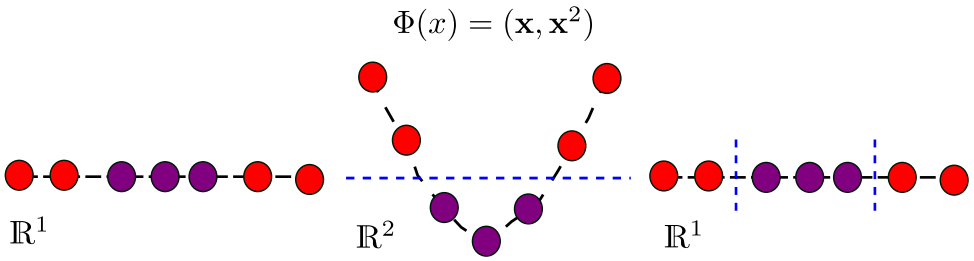


Figure 1.1.6: Φ mapping for a one-dimensional dataset where the problem becomes linearly separable.

so-called kernel trick [111], a procedure that maps the data into a higher-dimensional, or even infinite, feature space $\mathcal{H}$ via some mapping Φ. This allows the formulation of nonlinear variants of any algorithm that can be cast in terms of inner products between data points. These techniques have emerged from a combination of some research branches such as mathematical analysis, operations research and machine learning theory, and have spread far beyond the standard support vector machine algorithm in such a way that virtually each learning technique has been (or could be) redesigned to exploit the power of kernel functions. Indeed, this kernel function implicitly determines the feature space $\mathcal{H}$ in such a way that a poor choice of this function can lead to significantly impaired performance. These choices are related to the definition of a metric between input patterns that fosters correct classification. Usually, a parametrised set of kernels is considered for this purpose, although it is still necessary to choose a performance measure and an optimisation strategy. This optimisation is often performed using a grid-search or cross-validation procedure over a previously defined search space.

For the sake of understanding, consider the case of the Gaussian kernel function applied to a binary classification toy dataset, where one class is a circle and the other one is an ellipsoidal ring. The output of applying this function would be equivalent to the result represented in Figure 1.1.7 where a Gaussian distribution is considered for each pattern. The axes x and y are the ones associated to the data and z represents the sum of Gaussians centred in each point. This would be equivalent to introducing a similarity notion of the data, where within-class data is intended to be as similar as possible and vice versa.

Other learning strategies have also emerged apart from the above-mentioned and well-known cross-validation technique with the aim of better suiting a given dataset. These techniques are referred to as kernel learning strategies. Ideally, we would like to find the kernel that minimises the true risk of a specific classifier for a specific dataset. Unfortunately, this quantity is not accessible; therefore and as said, different estimates or bounds have been developed based on both analytical and experimental knowledge, such as the span of support vectors [109], the radius margin bound [111] or kernel-target alignment [29]. This problem has also been tackled using evolutionary algorithms [35, 54, 88],

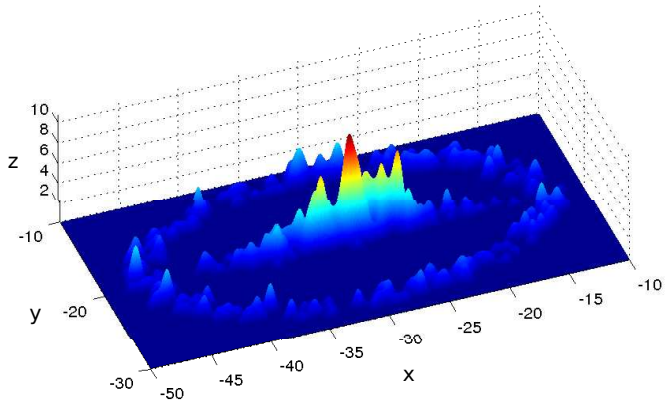


Figure 1.1.7: Representation of the sum of Gaussians centred in each point for an ellipsoidal ring toy dataset.

meta-learning approaches [98], or Bayesian inference [99], by defining data-dependent estimates of the complexity of a function class [66, 8] or simply by optimising the class separability in the feature space [115]. In most of these cases, a large amount of computation time is needed because the bounds or the algorithms require training the learning machine several times and might even require solving an additional optimisation problem. Moreover, some of the bounds are not differentiable, which means that they must be smoothed to use a gradient descent method [21], which can result in a loose solution for the problem that is tackled. A study of all these methods could be conducted in order to select the most appropriate one (in terms of accuracy, computational load, differentiability, etc.).

Kernel-target alignment [29] is a promising technique, where the kernel matrix is aligned to the ideal one by optimising its parameters. In this sense, Figure 1.1.8 shows the idea of selecting the closest kernel matrix $\mathbf{K}_{\boldsymbol{\alpha}}$ from a family of kernels $\mathcal{Q}$ to the ideal matrix $\mathbf{K}^*$, according to a distance relation $\mathcal{D}$. $\boldsymbol{\alpha}$ represents the parameters associated to the kernel and $\mathcal{K}$ the set of positive semidefinite kernels.

On the other hand, some authors suggest the use of the multi-scale kernel [21] (also known as a multi-parametric, anisotropic or ellipsoidal kernel), where a different kernel parameter is chosen for each feature. The general motivation for the use of multi-scale kernels is that, in real-world applications, the attributes can present very different natures, which hampers the performance of spherical kernels (i.e. with the same kernel width for each attribute). It is clear that more flexible kernels could fit heterogeneous datasets better, leading to a lower generalisation error [58, 40]. However, the number

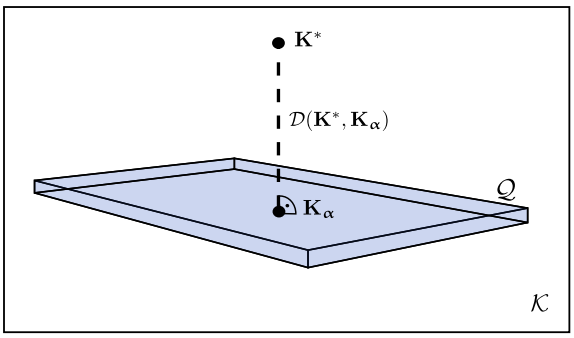


Figure 1.1.8: The most appropriate kernel for learning is $\mathbf{K}_{\boldsymbol{\alpha}}$ (the one nearest the ideal one ($\mathbf{K}^*$) according to some measure of similarity $\mathcal{D}$, being $\mathcal{K}$ the set of positive definite kernels).

of parameters of the kernel function (as many parameters as the number of features) makes the computational cost prohibitive when considering a cross-validation technique. For this reason, these kernels have been barely used in the literature; when they have been used, they have been optimised by evolutionary algorithms [45, 88, 40] or by gradient-based techniques applied to some measure of kernel quality or generalisation error bound [96, 21, 58]. The use of these kernels could be explored in order to analyse its potential.

Finally, it is also important for the topic of this thesis to define the notion of the empirical feature space [95, 116], which has been used for kernelising any given learning algorithm without needing to reformulate the method in terms of inner products. This space is Euclidean and preserves the geometrical structure of the original feature space, given that distances and angles in the feature space are uniquely determined by dot products and that the dot products of the corresponding images are the original kernel values. Note than when applying a nonlinear kernel which perfectly fits the data, the patterns in the empirical feature space will be linearly separable, which is an interesting feature for a wide range of uses. Figure 1.1.9 shows the relations between the input space, feature space $\mathcal{F}$ and empirical feature space $\mathcal{E}$. Note that φ is a linear operation.

1.1.3. Real-world problems

Several real-world problems are considered in this thesis in order to validate the methodologies proposed and solve some important difficulties which have been found for these applications.

Donor-recipient matching in liver transplantation

During the last few decades, new trends in biomedicine have considered using some machine learning techniques as classification methods [68, 117], which has worked well in a great number of problems and resulted in remarkable applications for science [105,

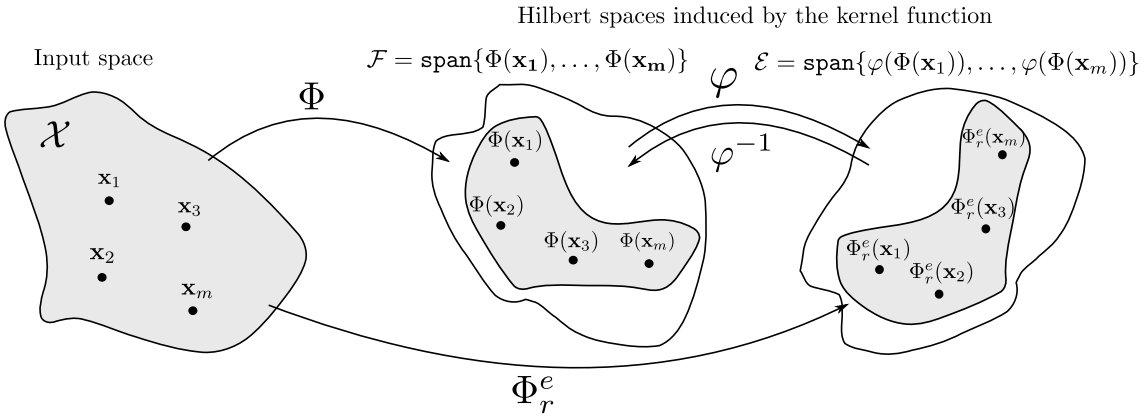


Figure 1.1.9: Representation of the relation and mapping between input space, feature space and empirical feature space.

100]. Liver transplantation is an accepted treatment for patients who present end-stage liver disease. However, transplantation is restricted by the lack of suitable liver donors; this imbalance between supply and demand resulting in significant waiting list death. In order to cope with this situation, several methods have been developed and applied to find a better system to prioritize recipients on the waiting list.

The first attempt at developing a system was the Donor Risk Index (DRI) [38], aimed at establishing the quantitative risk associated with the transplant when considering donor information. Another widely validated methodology that is the cornerstone of current allocation policy is the Model for End-stage Liver Disease (MELD) [64], which is based on the "sickest-first" principle, where the only aspect considered is information concerning the recipient. The use of expanded criteria donors (donors with extreme values of age, days in the intensive care unit (ICU), inotrope usage, body mass index (BMI) and cold ischemia time) results in an increased risk of recipient and/or graft losses compared to the risk associated with the use of livers from non-extended criteria donors [19]. These risks should be carefully analysed since the combination of several of these risk factors can result in graft loss [15]. Nevertheless, these methods can not be considered good predictors of graft failure after transplantation since they only take into consideration either characteristics of donors or of recipients (but not both), when there could actually be more complex factors involved in the situation (donor, recipient and transplant organ characteristics). In order to deal with this problem, Rana *et al.* [91] devised a scoring system (SOFT) that predicted recipient survival three months after liver transplantation, which is intended to complement MELD-predicted waiting list mortality rates by making use of both donor and recipient characteristics. P. Dutkowski *et al.* recently proposed a balance of risk (BAR) score [37] based on donor and recipient characteristics. A rule-based system was used to determine graft survival one year after the transplant [30, 14]. The input of this rules-based system being the response of two artificial neural networks

trained with donor, recipient and transplant organ characteristics.

Figure 1.1.10 graphically represents the process of organ allocation (figure restructured from [94]). Generally, donors are assigned to the candidates under the greatest-risk according to the MELD score. This policy does not allow the liver transplant team to match the donor to the recipient according to principles of fairness and benefit. This could lead to a risk of unconscious gaming when trying to match marginal donors to urgent candidates.

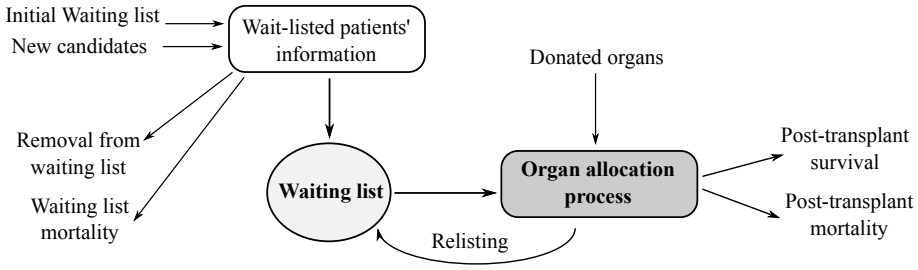


Figure 1.1.10: Graphic representing the organ allocation process.

A significant contribution in this area could be made by addressing the classification problem using an ordinal regression point of view since the classes could be designed taking into account the time leading up to liver failure (in case of failure) and, therefore, provide more thorough information than the common binary approach. The classes involved in the dataset could be: 1) failure of the graft within the first 15 days after transplantation, 2) failure between 15 days and 3 months, 3) failure between 3 months and one year, and 4) no failure presented. These intervals have been highlighted by experts as being the most pertinent in early graft loss. However, several issues need to be taken into account in order to exploit the presence of this order structure. First of all, the learner (classifier, in this case) could benefit from this implicit ordering in order to construct more robust and fairer decision regions for the data, since the classification errors to be minimised vary from the ones considered in the nominal classification paradigm (the zero-one loss function). Secondly, with the final aim of evaluating the performance of those classifiers, different measures or metrics could be developed and used. Other factor to consider when approaching this problem is the imbalanced nature of the problem, which should not be ignored in order to derive non-trivial and fair classification models.

Analysis of sustainable development

Sustainable development (SD) is among the most relevant and pressing challenges of the modern age. Since the work of Meadows *et al.* [85], the interest in this problem has been increasingly growing in the political arena and social consciousness. The underlying idea still remains: human consumption is outstripping what the planet can produce, as we are spending natural resources faster than they can be replenished. In this sense,

the academic community, main stakeholders and the political and media debate display special interest in achieving SD as a model of growth for nations and as a primary goal. In times of a deep economic crisis, society and policy-makers focus their attention on economic indicators such as income and employment rates; however, sustainability and social inclusion should also be a priority.

Usually, what is commonly established is that SD is concerned with ensuring long-term human well-being, which necessarily involves confronting the challenges of limited natural resources and global poverty, having a good standard of living, a long and healthy life, access to education, participation in the social and political life of the community and well-paid work that provides people with the opportunities to achieve their goals, hopes and aspirations [107].

The imperious need for reliable and pertinent indicators, to better monitor and foster SD and to guide this SD process at a national level, was recognised early at the time of the Rio Conference and the Agenda 21 [106], followed by the Commission on Sustainable Development work programme on indicators. The most common effect of indicators could be calling attention to an existing problem. However, these indicators yield different scores and rankings depending on the nature and type of assessments. They also report on past performance [3] and they do not predict whether a certain country is heading (or could head) a group in these terms.

A great deal of measurement attempts have been developed over the last two decades at various levels (international organisations, academic and private initiatives) for managing and monitoring progress towards SD [13, 32, 90, 72, 89], some of which have focused on households, distribution of wealth, quality of life, social progress and ecological sustainability [4, 48, 59], but without a consensus on which one is the most determinant at a general level [70].

Among these initiatives, actions to improve and complement the current growth measurements [59] are frequently outlined. This is important, as there are various dimensions that can be interlinked, e.g. education or employment quality can affect health, social relations and status, civic participation, etc. The motivation for proposing an alternative methodology to composite indicators or indices could be that, in the first place, indices summarise too much and provide less information than the description of the characteristics of a cluster or the analysis of models able to predict the class for a new pattern. On the other hand, these indices have been found to be very sensitive to the choices of the index's construction and the selection of the variables to be used.

Machine learning could be, in this sense, very helpful. For example, it could give us some clues about how to group the chosen countries according to different indices associated to their sustainable development (always supervised by an expert). Ordinal

classification could be also used to learn an interpretable model to rank these countries which in the future could be used for for monitoring the progress towards SD of the different countries and to extract some knowledge of the most significant characteristics to measure this progress.

The Unequal Area Facility Layout Problem

Facility layout design (FLD) determines the placement of facilities in a manufacturing plant with the aim of deciding the most effective arrangement in accordance with some criteria or objectives, under certain constraints. In this respect, Kouvelis *et al.* [69] outlined that FLD is very important for production efficiency because it directly affects manufacturing costs, lead times, work in process and productivity. According to [104], well laid out facilities contribute to the overall efficiency of operations and can reduce between 20 % and 50 % of the total operating costs. There are many kinds of layout problems. In short, the unequal area facility layout problem (UA-FLP) considers a rectangular plant layout that is made up of unequal rectangular facilities that have to be placed effectively in the plant layout (as can be seen in Figure 1.1.11, where the different facilities, the material flow and some restrictions are represented in different colours).

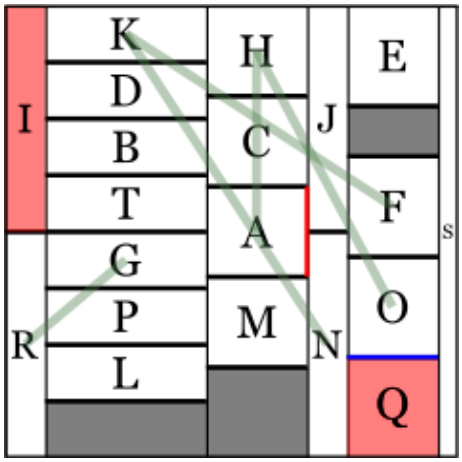


Figure 1.1.11: Facility layout example.

Most authors have solved this problem using quantitative criteria. Unfortunately, the approaches may not adequately consider all the essential qualitative information that affects a human expert involved in design (e.g. engineers, stakeholders, regulators, etc.) [6]. In this way, qualitative features are also important to be taken into account, such as location preferences of certain facilities, the way that remaining space is distributed in the layout or any other subjective consideration that can be judged as relevant for the decision maker (DM). Besides, these features can be subjective, unknown a priori and changing during the procedure evolution. So that, it is difficult to take into account both quantitative

and qualitative aspects at the same time, because they can not be easily formulated as an objective function. Consequently, including the expert knowledge is vital to incorporate qualitative considerations in the design. This fact also gives us the following benefits: finding a solution that is satisfactory to the DM (but that it is not necessarily an optimal solution) [5, 76], choosing the best trade-off solution when a conflict among objectives or constraints exists [62], assisting the algorithm in achieving the search process towards the DM preferences [80, 22], excluding the need to give all the required preference information a priori, offering the DM the possibility to know about his/her own preferences [62], stimulating the creativity of the DM [93] and getting original and feasible designs.

Many evolutionary computation approaches have been applied to UA-FLP. Among these, the genetic algorithms (GAs) [53] are frequently considered. Brintup *et al.* [17] have emphasised that interactive evolutionary computation can greatly help improve design by involving experts in searching for a satisfactory solution [16].

An interactive genetic algorithm (IGA) previously developed for FLD [44] considers user evaluation and handle subjective features. This algorithm uses a clustering mechanism to reduce the number of evaluations required from the user. However, running the IGA can be a tedious task for a designer, as many evaluations are still required. Fatigue is the main reason for an early stop of IGAs, thus reducing the possibilities of the system to find better designs. Moreover, user evaluation is some orders of magnitude slower than computed evaluation, leading to a much smaller search capacity. Learning user design preferences over a concrete layout problem would allow to simulate user responses. In this way, fatigue could be avoided and search could be performed much faster, which is specially useful in the context of the large search space of facility layouts. An interesting contribution would be then to design a system able to learn these user layout preferences and perform a search considering both, the user preferences and other objective criteria (it is clear that such a learning of the user preferences should be performed by means of ordinal classification). In this vein, Figure 1.1.12 shows the process to generate a multi-objective optimisation algorithm that simulates the expert opinion in order to explore the search space in depth.

States of development in fish oocytes

The analysis of microscopic images of fish gonad cells (*oocytes*) is a useful tool to estimate parameters of fish reproductive ecology and to analyse fish population dynamics. The assessment of oocyte development dynamic and fecundity is a fundamental topic in the study of reproductive biology and population dynamics [57]. To estimate fecundity with accuracy, only mature oocytes must be considered, which requires a reliable classification of oocytes according to its state of development. The best method to classify

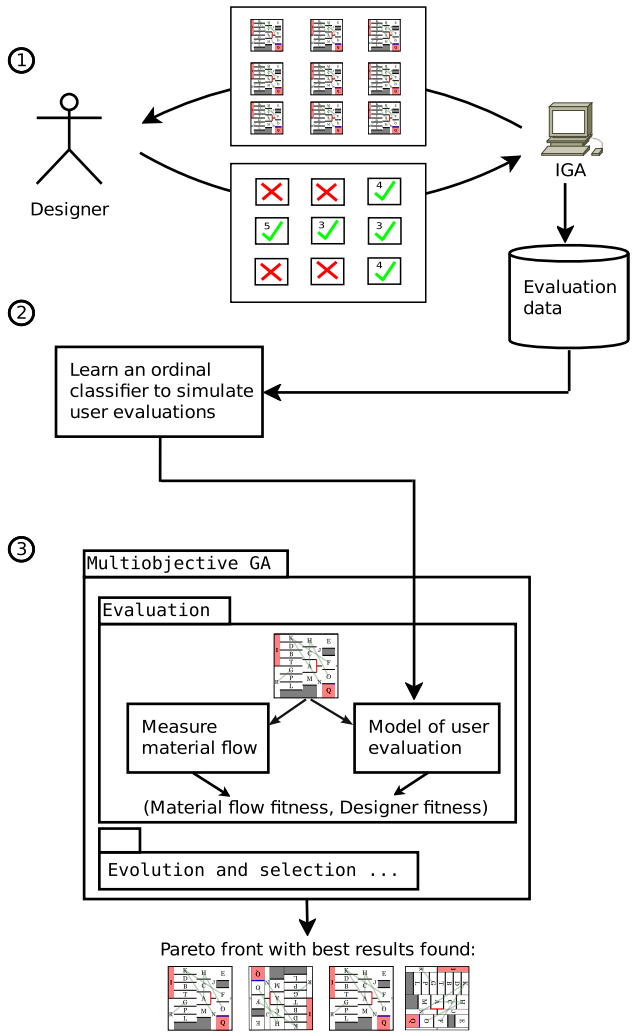


Figure 1.1.12: Diagram for a facility layout search system.

oocytes is histology, although experienced personnel is required. The main developmental states of oocytes are: *Primary Growth* (PG), *Cortical Alveoli* (CA), *Vitellogenic* (VIT), *Hydrated* (HYD) and *Atretic* (AT). The PG state corresponds to immature oocytes; CA, VIT and HYD to mature ones; and AT corresponds to those mature oocytes that will be resorpted (i.e. non-ovulated). Depending on the objective of the study, these main states could be divided in sub-states. Specifically, the specie *Reinhardtius hippoglossoides*, also known as Greenland halibut, presents some irregularities in the maturation processes [86, 63] that could suggest that individual spawning does not necessarily occur on an annual basis as for most exploited fish. This specie presents a unique reproductive development pattern, with ovaries simultaneously containing oocytes developing for the current and subsequent reproductive seasons [92, 65]. Four sub-levels of development within the VIT state have been identified (VIT1, VIT2, VIT3 and VIT4) in this specie (see Figure 1.1.13). When maturation begins, a group of oocytes evolves from PG to CA and progresses until reach VIT2;

then some oocytes (called the leading cohort) continue the progression (VIT3-VIT4-HYD), while the rest of mature oocytes (secondary cohort) remains in VIT2 (likely until the next spawning season) or become AT. To analyse oocyte cohort dynamic and estimate egg production it is necessary to classify correctly the VIT sub-states (note that these sub-states also present an ordinal nature, therefore motivating the application of such techniques in this case).

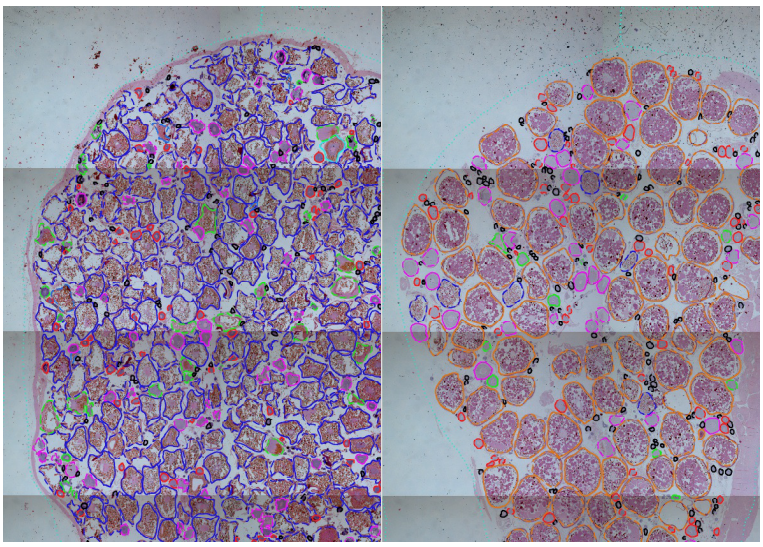


Figure 1.1.13: Examples of histological images of fish specie *Reinhardtius hippoglossoides*. The cell outlines were manually annotated by experts using the Govocitos software tool. The color identifies the state of development of the oocyte: black (PG), red (CA), pink (VIT1), cyan (VIT2), blue (VIT3), orange and green (VIT4).

1.2. Motivation and challenges

From the previous section, two main issues can be noted concerning ordinal classification: firstly, the need to define and use different metrics for error characterisation and, secondly, the exploitation of the data ordering. These two issues have been the focus of most of the ordinal classification works of the related literature. However, in addition to these, there are other issues that have not been solved up to the date. Firstly, since class labels often represent ranks or degrees, class overlapping is common between all or a subset of classes. Specially for latent variable models, which are by far the most extended methods [112, 47], high dimensionality and nonlinearly separable data can make the mapping function result into complex models that impose highly nonlinear transformations from the input space to the latent space. Imposing too rigid models in these projections can derive into problems for classifying patterns in the class boundaries, specially in the presence of the abovementioned noise or class overlapping. Secondly, as said, the most used and

successful methods for this learning setting are threshold methods, i.e. models that try to discover a mapping function $f(\mathbf{x})$. Usually, the optimisation of a nonlinear separation function is much more difficult than when dealing with linear functions, unless the kernel trick is used (which is the case for most of ordinal classification algorithms). However, kernel learning techniques specially designed for these algorithms have not been derived yet and neither does a general method to kernelise any ordinal regression algorithm (independently of whether it can be cast in terms of inner products between patterns). Finally, the class imbalance problem (a problem that poses a serious hindrance for learning and which is encountered when one or several categories present a much lower prior probability) is also present in ordinal classification problems [56, 31, 47]. Indeed, for most real world ordinal datasets, there are some classes that are naturally much more probable than others (specially when the number of classes is high) and therefore the problem presents an imbalanced nature.

Considering the ahead issues, we can synthesise the following open challenges that constitute the objectives of this thesis:

State-of-the-art in ordinal regression. Compared with nominal classification, ordinal regression is a machine learning field much less studied and explored. However, there are several works and literature dealing with ordinal regression, and it is necessary to properly analyse them. A taxonomy of ordinal regression methods and an effort to gather and compare all of the algorithms would help to contextualise the proposals in this field.

Ordinal decomposition methods and a more general technique. As will be analysed, one of the most simple and common approaches to ordinal regression is to decompose the labelling space into more simpler ones and solve the associated classification problems separately in order to finally merge all of the output responses into a single output variable. This formulation allows the use of binary classification methods and usually obtains good results at a generally reasonable computational load. This thesis will analyse the different proposals under this framework in order to formulate a more general technique.

Ordinal kernel learning techniques. Kernel methods are one of the most common choices when performing an ordinal regression learning task. There methods help to deal with the nonlinear nature of the data and have shown to perform very well for this learning setting as opposed to other pioneer linear methods. However, few efforts have been made to reformulate kernel learning algorithms and create a general method to kernelise any ordinal regression algorithm (which could be useful to improve the abovementioned linear methods).

Class imbalance. Classification methods in machine learning often conveniently assume that the prior class probability distribution is of high entropy. However, this is not the case in many real-world applications from areas such as medical diagnosis, information retrieval, fraud detection, fault monitoring and others. The classification paradigm when one or several classes have a much lower prior probability is known as imbalanced classification and poses a serious hindrance for the learning process. Imbalanced datasets are very common in the setting of ordinal regression, because there are some classes that are, by nature, of lower probability. Therefore, the study of the existing sampling methods is interesting in order to reformulate the most adequate ones for the paradigm of ordinal regression, and test then their performance.

The analysis of different potential real-world applications. In this thesis, the applicability of the different developed techniques is also going to be tested in order to analyse their performance in a different framework than those of benchmark datasets. To do so, different potential fields of research in which ordinal learning problems could be considered will be selected and studied, such as the ones discussed in previous section.

1.3. Objectives

The present thesis addresses the aforementioned open challenges. All these challenges result in the following formal objectives.

1. State-of-the-art in ordinal regression objectives:

   a) *To propose an ordinal regression method taxonomy*. The first part in the study consist on a review of the state-of-the-art in ordinal regression. Although this objective is implicit in most theses, ordinal regression is a very recent field and, up to the authors knowledge, there are not surveys related to this topic. Thereafter, it is even more important to collect related works and also to propose a taxonomy to organize existing methods in order to properly contribute to the state-of-the-art.

   b) *To select benchmark datasets*. Preliminary exploration of the state-of-the-art suggests that there are no public specific datasets repositories for ordinal classification. The most used dataset repository in the literature is the *ordinal regression benchmark datasets* provided by Chu et al. [25]. However, the benchmark datasets provided by Chu and Ghahramani are not real ordinal classification datasets but regression ones for which the target variable has been discretised

into several equal-width bins. We identify two problems regarding these datasets: first, as said, the datasets are not real classification datasets; and second, using same width intervals for classes label generation produce an artificial class imbalance in the datasets.

c) *To test their performance in real-world data*. Most of the developed methods in this setting have been tested using the aforementioned regression datasets, being their performance not realistic. Therefore, it could be useful to study and compare ordinal and nominal methods in a real-world application in order to analyse the potential improvement in the results.

2. The study and development of a decomposition method, which can be divided in the following objectives:

   a) *To derive a general method to decompose the labelling space* considering the ordinal nature of the data and focusing equally on each class in order to take into account the potential imbalance of the data.

   b) *To propose and test different approaches for merging the predictions from the classifiers of the different decomposed learning problems*. More specifically, compare static approaches to that ones that require a specific training (dynamic approaches).

3. The following objectives try to face the kernel learning open challenge:

   a) *To analyse most common and successful techniques for kernel learning*. To enumerate the advantages and disadvantages of these methods in order to analyse which will is the best suited to be redefined for tackling ordinal regression problems.

   b) *To reformulate and improve the algorithm chosen in the previous objective to ordinal regression* and test whether it achieves significant better performance than other chosen ordinal kernel methods.

   c) *To derive a method to kernelise any linear ordinal regression algorithm* (even those that can not be cast in terms of dot products) in order to improve its performance and compare it to other learning methods which can deal with nonlinear data boundaries (and therefore study whether the detriment in the results of these linear methods as compared to others was only produced by their inability to properly exploit nonlinear data).

4. The class imbalance challenge has resulted in the following objectives:

a) *To review the different alternatives that have been considered in the imbalanced classification literature* in order to properly identify the most successful techniques and solve the different issues that have been noted in some of the nominal approaches.

b) *To address these issues* and formulate a more general and flexible methodology for this problem (in this case, for nominal classification).

c) *To explore new solutions considering ordinal imbalanced problems* in order to analyse their behaviour. More specifically, ordinal over-sampling methods are to be derived not only to balance the class distribution but also to improve the quality of data.

5. Application of ordinal regression methods to real-world problems. In this thesis, some learning problems will be addressed under the umbrella of ordinal regression. The ultimate goal is to justify the research in this field. There are three objectives associated to this open challenge:

   a) *To develop a model for donor-recipient matching in liver transplantation*. Learning models are being developed nowadays for the complex task of assigning the most compatible recipient to a donor in organ transplantation based on the survival principle. The main problem of these methods is that they only consider the binary classification task (success or potential rejection of the organ), not allowing the discrimination between different patients which are predicted to be compatible. We will propose a more fine model, which, ideally, will be capable of predicting the time leading up to organ failure (if it does indeed occur) by using an ordinal regression approach. This thesis aims at testing and analysing this model in order to extract some knowledge of its behaviour.

   b) *To develop a tool to rank the EU countries according to their advances in sustainable development*. Sustainable development is a major challenge for nations, even more so in the current economic crisis and uncertain environment. Although different indicators, indices and rankings to measure and monitor these advances at the macro level exist, the benefits for stakeholders and policy makers are still limited because of the absence of predictive models. In order to do so and to analyse the characteristics that could be said to be of most importance for sustainable development ordinal methods will be also used.

   c) *To analyse the usefulness of ordinal classification in other real-world settings*. The other two applications presented (the facility layout problem and the classification of states of development in occytes) will be considered in this case in order to study the potential superiority of ordinal classifiers over nominal ones. The

case of the facility layout is of special interest, as it will demonstrate whether ordinal classification should be used when considering user preferences.

1.4. Summary of the thesis

The concept of machine learning is one of the most actively researched in the branch of artificial intelligence. The most basic idea behind this paradigm is to create intelligent models able to automatically learn from data. The applicability of machine learning is vast, given its naturally generic root (i.e. discovering patterns from data). Therefore, some application examples could be found in areas such as robotics, medicine, finances, microbiology, agronomy and others, where a predictive or data clustering model could be of vital importance for some specific purpose.

Within this machine learning research area, different paradigms can be found. However, the classification paradigm is one of the most useful and studied ones given its relevance. This paradigm can be subsequently divided into several groups, according to the nature of the different types of models and the considered data. Each subdivision of machine learning is focused on different challenges and aims, and their study is of great significance for the literature and for its applications. Ordinal classification (also called ordinal regression) is one of these examples. It is a less studied area with great potential and a wide range of possible and helpful applications. The main topic of this thesis is precisely the study of the ordinal classification paradigm: the discovery and identification of the different challenges, the study of the existing approaches and the development of new techniques. The main difference between this paradigm and the standard classification one is the natural order between the categories or classes in the problem. In contrast to regression, the number of ranks is finite and there is no knowledge about the distances among them. Ordinal classification problems are common in real-world, a few examples being preference learning, credit rating or risk rating.

There are several challenges that have been identified for the concept of ordinal classification and that form the research line of the present thesis. More specifically, these are: the necessity of reviewing the state-of-the-art, exploring in depth decomposition techniques, deriving specific kernel learning techniques and approaching the problem of imbalanced classification.

As stated before, ordinal classification has been less explored when compared to binary or multinomial classification. However, several works in the literature have been published, making therefore their analysis a necessary and preliminary step of the current thesis. A taxonomy of ordinal regression methods and an effort to gather and compare the main methods, datasets, libraries and metrics for their evaluation would help to con-

textualise the proposals in the field. This is precisely the first challenge of the current thesis. In this sense, the literature regarding ordinal regression has been reviewed and the methods have been organised according to a proposed taxonomy. The main methods for each group have been selected and their performance is compared in a wide and thorough statistical analysis. Different techniques outstand from these groups depending on different criteria, such as time or several evaluation metrics. Moreover, the proposals are studied on a real-world database in order to validate the potential advantages of ordinal methods over nominal techniques in practice. More specifically, the case of predicting the state-of-development of occytes is considered.

As it is noted in this review of the state-of-the-art, one of the most common approaches to ordinal classification is the use of a labelling decomposition procedure. This formulation allows the use of binary classification methods and has been shown to lead, in general, to good results at a reasonable computational load. This thesis analyses the different proposals under this framework in order to formulate a more general ordinal technique. Different strategies for fusing the outputs of the constructed models are explored (such as the error correcting output codes or different trainable strategies), as well as the use of decomposition methods for imbalanced classification problems (via the use of a cascade utility model). The interpretability of such decomposition approaches is also analysed when using linear models. Two applications are taken into account in this case: the rating towards sustainable development of EU countries and the construction of a donor-recipient assignation model in organ transplantation.

Moreover, kernel mapping is one of the most widespread approaches to intrinsically derive nonlinear classifiers. However, few efforts have been paid to study and compare the different kernel learning techniques proposed in the literature, i.e. methods focused on selecting and adjusting the kernel function to the data itself. Each of these methods is designed for a very specific setting or data distribution. Then, the analysis of these techniques is vital to highlight the most advantageous ones. Kernel methods are also a common choice for ordinal classification algorithms, since these methods help to deal with the nonlinear nature of the data and have shown to perform very well for this learning setting, as opposed to other pioneer linear methods. However, no kernel learning technique has been derived until the date for ordinal classification. This thesis introduces a first approximation to this idea, where a previous analysis of kernel learning techniques is also conducted, in order to highlight the most interesting ones. Furthermore, the empirical feature space has been used for kernelising several binary classification algorithms that can not be cast in terms of inner products. We propose the use of this empirical feature space for obtaining a nonlinear version of the most commonly used ordinal regression linear classifiers. A dimensionality reduction technique is usually employed, in order to reduce the data dimensionality while approximating the data. This proposal is also extended to the case of

ordinal classification by maintaining the ordinal information when reducing the number of dimensions.

Finally, the imbalanced nature of some real-world data is also one of the current challenges for machine learning researchers. Most approaches to deal with this issue oversample the minority class through convex combination of its patterns, which could result in data inconsistencies. The nature of these inconsistencies is explored and they are fixed in this thesis in order to provide a more robust and fair method for tackling the data imbalance. To do so, the notion of empirical feature space induced by a kernel function, which has been mentioned before, is used. Class imbalance has also been seen to represent a general issue for ordinal classifiers, since there are classes, usually the most extreme ones, which are naturally of lower probability. The study of a method, then, able to consider the order information when generating new synthetic patters is considered by means of graph-based techniques.

Note that all of the above-mentioned challenges and hypothesis have been validated by using a set of benchmark ordinal classification and binary datasets (when necessary) and thorough statistical comparisons. The majority of the datasets are extracted from the well-known UCI repository. In addition, some of the ordinal regression benchmark datasets were also considered, since they are widely used in the ordinal regression literature.

The work done in this thesis is reflected in thirteen papers in international conferences and nine papers in international journals.

1.5. Publications

Figure 1.5.1 shows a diagram presenting the topic (or topics) related to each of the publications associated to the current thesis. The following international journal papers include some of the ideas of this thesis:

J1 M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás Martínez. Projection-Based Ensemble Learning for Ordinal Regression. *IEEE Transactions on Cybernetics*, 44(5):681–694, 2014, Impact Factor (2013): 3.781 (Q1).

J2 M. Pérez-Ortiz, M. Cruz-Ramírez, M. D. Ayllón-Terán, N. Heaton, R. Ciria and C. Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. *Applied Soft Computing*, 14:88–98, 2014, Impact Factor (2013): 2.679 (Q1).

J3 M. Pérez-Ortiz, M. de la Paz-Marín, P.A. Gutiérrez and C. Hervás-Martínez. Classification of EU countries' progress towards sustainable development based on ordinal

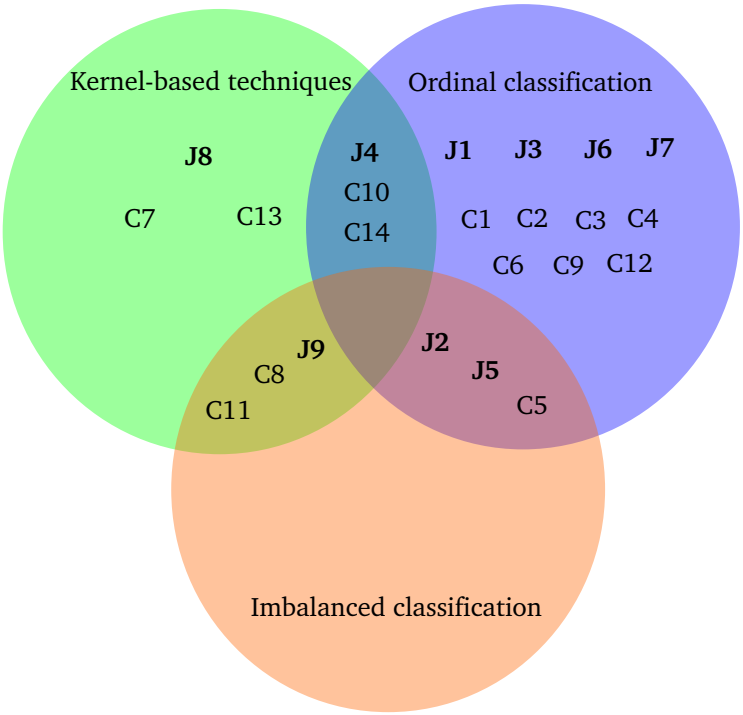


Figure 1.5.1: Diagram for the publications associated to this thesis and their relation to the three main topics of the thesis: ordinal classification, kernel-based learning and imbalanced data.

regression techniques. *Knowledge-Based Systems*, 66:178–189, 2014, Impact Factor (2013): 3.058 (Q1).

J4 M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelising the Proportional Odds Model through Kernel Learning techniques. *Neurocomputing*, In press, 2014, Impact Factor (2013): 2.005 (Q1).

J5 M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez and X. Yao. Graph-Based Approaches for Over-sampling in the context of Ordinal Regression. *IEEE Transactions on Knowledge and Data Engineering* (TKDE), In press, 2015, Impact Factor (2013): 1.815 (Q1).

Some ideas have been submitted to different journals and are currently under review process:

J6 P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro and C. Hervás-Martínez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering* (Second Revision), 2014, Impact Factor (2013): 1.815 (Q1).

J7 M. Pérez-Ortiz, M. Fernández-Delgado, E. Cernadas, R. Domínguez-Pétit, P.A. Gutiérrez and C. Hervás-Martínez. On the use of nominal and ordinal classifiers for

the discrimination of states of development in fish oocytes. *Neural Processing Letters* (Second Revision), 2014, Impact Factor (2013): 1.237 (Q2).

J8 M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. A study on multi-scale kernel optimisation via centred kernel-target alignment. *Neural Processing Letters* (Under Review), 2014, Impact Factor (2013): 1.237 (Q2).

J9 M. Pérez-Ortiz, P.A. Gutiérrez, P. Tino and C. Hervás-Martínez. Over-sampling the minority class in the feature space. Submitted to *IEEE Transactions on Neural Networks and Learning Systems* (Under Review), 2014, Impact Factor (2013): 4.370 (Q1).

Moreover, some works have also been published in national and international conferences:

C1 M. Pérez-Ortiz, P.A. Gutiérrez, C. García-Alonso, L. Salvador-Carulla, J.A. Salinas-Pérez y C. Hervás-Martínez. Ordinal classification of depression spatial hot-spots of prevalence. In *Intelligent Systems Design and Applications* (ISDA), pages 1170–1175, 2011, Córdoba, Spain.

C2 P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez. An experimental study of different ordinal regression methods and measures. In *7th International Conference on Hybrid Artificial Intelligence Systems*, pages 296–307, 2012, Salamanca, Spain.

C3 M. Pérez-Ortiz, L. García-Hernández, L. Salas-Morera, C.Hervás-Martínez and A. Arauzo-Azofra. An ordinal regression approach for the unequal area facility layout problem. In *Soft Computing Models in Industrial and Environmental Applications* (SOCO), pages 13–21, 2012, Ostrava, Czech Republic.

C4 M. Pérez-Ortiz, A. Arauzo-Azofra, C. Hervás-Martínez, L. García-Hernández y L. Salas-Morera. A system learning user preferences for multiobjective optimization of facility layouts. In Soft Computing Models in Industrial and Environmental Applications (SOCO), pages 43–52, 2012, Ostrava, Czech Republic.

C5 M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, J. Briceño y M. de la Mata. An ensemble approach for ordinal threshold models applied to liver transplantation. In *International Joint Conference on Neural Networks* (IJCNN), pages 2795–2802, 2012, Brisbane, Australia.

C6 P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, and C. Hervás-Martínez. Estudio comparativo de distintos métodos de umbral en regresión ordinal. In *IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)*, pages 872–881, 2013, Madrid, Spain.

C7 M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. Multiscale support vector machine optimization by kernel-target alignment. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), pages 391–396, 2013, Bruges, Belgium.

C8 M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Synthetic over-sampling in the empirical feature space. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), pages 385–390, 2013, Bruges, Belgium.

C9 M. Pérez-Ortiz, R. Colmenarejo, J.C. Fernández-Caballero y C. Hervás-Martínez. Can machine learning techniques help to improve the common fisheries policy?. In *International Work Conference on Artificial Neural Networks* (IWANN), Lecture Notes in Computer Science Volume 7903, pages 278–286, 2013, Tenerife, Spain.

C10 M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelizing the proportional odds model through the empirical kernel mapping. In *International Work Conference on Artificial Neural Networks* (IWANN), Lectures Notes in Computer Science Volume 7902, pages 270–280, 2013, Tenerife, Spain.

C11 M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Borderline kernel based oversampling. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8073, pages 472–481, 2013, Salamanca, Spain.

C12 M. Pérez-Ortiz, P.A. Gutiérrez y C. Hervás-Martínez. Log-gamma distribution optimisation via maximum likelihood for ordered probability estimates. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8480, pages 454–465, 2014, Salamanca, Spain.

C13 M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Learning kernel label decompositions for ordinal classification problems. In *International Conference on Neural Computation Theory and Applications*, pages 218–225, 2014, Rome, Italy.

C14 M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Incorporating privileged information to improve manifold ordinal regression. In *International Conference on Neural Computation Theory and Applications*, pages 187–194, 2014, Rome, Italy.

Other publications done during the PhD.:

- M. Cruz-Ramírez, C. Hervás-Martínez, P.A. Gutiérrez, M. Pérez-Ortiz, J. Briceño y M. de la Mata. Memetic pareto differential evolutionary neural network for donor-recipient matching in liver transplantation. *Soft Computing*, 17(2):275–284, 2013, Impact Factor (2013): 1.304 (Q2).

- L. García-Hernández, M. Pérez-Ortiz, A. Arauzo-Azofra, L. Salas-Morera y C. Hervás-Martínez. An evolutionary neural system for incorporating human expert knowledge into the UA-FLP. *Neurocomputing*, 135:69–78, Impact Factor (2013): 2.005 (Q1).

- J. Sánchez-Monedero, P.A. Gutiérrez, M. Pérez-Ortiz y C. Hervás-Martínez. A n-spheres based synthetic data generator for supervised classification. In *International Work Conference on Artificial Neural Networks* (IWANN), Lectures Notes in Computer Science Volume 7902, pages 613–621, 2013, Tenerife, Spain.

- M.D. Ayllón, R. Ciria, R. Valente, M. Cruz-Ramírez, M. Pérez-Ortiz, C. Hervás-Martínez, M. Rela, J. O'Grady, M. de la Mata, N.D. Heaton, J. Briceño. External european validation of a multicenter model for donor-recipient matching in liver transplantation based on artificial neural networks. In *2014 International Liver Congress*, pages S381, 2014, London, United Kingdom.

- M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero, C. Hervás-Martínez, Athanasia Nikolaou, Isabelle Dicaire y F. Fernández-Navarro. Time series segmentation of paleoclimate tipping points by an evolutionary algorithm. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8480, pages 318–329, 2014, Salamanca Spain.

- M. Cruz-Ramírez, M. de la Paz-Marín, M. Pérez-Ortiz y C. Hervás-Martínez. Time series segmentation and statistical characterisation of the Spanish stock market Ibex-35 index. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8480, pages 74–85, 2014, Salamanca, Spain.

- M. Rodríguez-Perálvarez, M. Cruz-Ramírez, E. Tsochatzis, C. García-Caparrós, P. A. Gutiérrez, G. Pieri, M. Pérez-Ortiz, D. Patch, J. L. Montero, J. O. Beirne, J. Briceño, A.K. Burroughs, C. Hervás-Martínez y M. de la Mata. The role of machine learning classifiers to predict early recurrence of hepatocellular carcinoma after liver transplantation, In *The International Liver Congress*, 2014, London, United Kingdom.

Learning is always rebellion... Every bit of new truth discovered is revolutionary to what was believed before.

Margaret Lee Runbeck

2

The state-of-the-art in ordinal regression

This chapter presents a review of the state-of-the-art in ordinal regression, including a taxonomy based on how the models are designed to take the order into account. Up to the authors knowledge there are not similar reviews in this field. Moreover, the chapter also includes a study of these classifiers on two real-world problems.

Main publications associated to this chapter:

- P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro and C. Hervás-Martínez. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering* (Second Revision), 2014, Impact Factor (2013): 1.815 (Q2).
- M. Pérez-Ortiz, M. Fernández-Delgado, E. Cernadas, R. Domínguez-Pétit, P.A. Gutiérrez and C. Hervás-Martínez. On the use of nominal and ordinal classifiers for the discrimination of states of development in fish oocytes. *Neural Processing Letters* (Second Revision), 2014, Impact Factor (2013): 1.237 (Q2).
- M. Pérez-Ortiz, A. Arauzo-Azofra, C. Hervás-Martínez, L. García-Hernández y L. Salas-Morera. A system learning user preferences for multiobjective optimization of facility layouts. In Soft Computing Models in Industrial and Environmental Applications (SOCO), pages 43–52, 2012.

Other publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez, C. García-Alonso, L. Salvador-Carulla, J.A. Salinas-Pérez y C. Hervás-Martínez. Ordinal classification of depression spatial hot-spots of prevalence. In *Intelligent Systems Design and Applications* (ISDA), pages 1170–1175, 2011.

- P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez. An experimental study of different ordinal regression methods and measures. In *7th International Conference on Hybrid Artificial Intelligence Systems*, pages 296–307, 2012.

- M. Pérez-Ortiz, L. García-Hernández, L. Salas-Morera, C.Hervás-Martínez and A. Arauzo-Azofra. An ordinal regression approach for the unequal area facility layout problem. In *Soft Computing Models in Industrial and Environmental Applications* (SOCO), pages 13–21, 2012.

- P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, and C. Hervás-Martínez. Estudio comparativo de distintos métodos de umbral en regresión ordinal. In *IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)*, pages 872–881, 2013.

- M. Pérez-Ortiz, R. Colmenarejo, J.C. Fernández-Caballero y C. Hervás-Martínez. Can machine learning techniques help to improve the common fisheries policy?. In *International Work Conference on Artificial Neural Networks* (IWANN), Lecture Notes in Computer Science Volume 7903, pages 278–286, 2013.

The three main publications are now presented in the different subsections of this chapter.

2.1. Ordinal regression: survey and experimental study

The following paper presents a study on ordinal classification methods, one of the current open challenges for machine learning researchers. In this sense, ordinal classification, or ordinal regression, focus on machine learning problems where there exist a natural order of the categorical variable. Many real-world applications present this structure and this has increased the number of methods and algorithms developed over the last years in this field. Therefore, it is interesting the study and analysis of the different approaches already published. Because of this, this paper aims at surveying the state-of-the-art and proposing a taxonomy for the different models. A thorough experimental design is conducted to test the difference of ordinal and standard multiclass methods when using ordinal data.

More specifically, in this paper, the problem tackled is defined formally and extensively to help the reader, and ordinal classification is distinguished from other paradigms, such as ranking, sorting or monotonic classification, among others. The taxonomy proposed mainly discriminates between naïve approaches (which are indeed the most common ones) and methods specially designed for ordinal classification. Under the term of naïve approaches several strategies can be found: the application of multiclass classification, regression or cost-sensitive techniques. The taxonomy also differentiates between ordinal classification methods, dividing these into the following categories: binary decomposition methods (distinguishing also between multiple model and single model approaches), threshold models (again separating discriminative, generative and ensemble models) and finally, augmented binary classification techniques.

The most representative proposals from the different families of the previous taxonomy are used for the experiments. 27 ordinal datasets are considered, as well as three evaluation metrics (the well-known accuracy, Acc, mean absolute error or MAE measure which is the most widely used metric in ordinal classification, and the time in seconds). Most datasets are publicly available real ordinal classification datasets extracted from benchmark repositories (UCI and mldata.org). However, other datasets representing regression tasks are also used, as they belong to one of the most widely used dataset repository for ordinal classification (the one provided by Chu et al. [25]). As previously discussed, these datasets are not real ordinal classification ones but regression problems, which are turned into ordinal classification, the target variable being discretised into K different bins with equal frequency.

Some conclusions are extracted from the results obtained in this paper. Firstly, it should be noted that several ordinal regression methods could be emphasised according to the different metrics considered. However, there are many factors that can influence the choice of the method, and all of them should be taken into account. It is also important to note the poor performance obtained by most pioneer linear methods, such as the proportional odds model [83], which is surprising given that it is one of the most widely used in real-world applications. The number of classes in the problem has been also seen to be an important factor when considering the scalability of the different methods, decomposition methods [39, 113] being the ones more affected by this. Both binary decomposition methods and threshold models (specially the reformulation of the support vector machine [26]) obtain outstanding results, as well as the augmented binary decomposition approach [74], that produces a remarkable trade-off between Acc and MAE. On the other hand, naïve approaches have been seen to obtain promising results for Acc but not for MAE, which is the main objective of this type of classification. Finally, if dealing with large- scale datasets, the most appropriate option is the use of the extreme learning machine algorithm for ordinal regression [33].

Ordinal regression methods: survey and experimental study

Pedro Antonio Gutiérrez, *Member, IEEE,* María Pérez-Ortiz, Javier Sánchez-Monedero, Francisco Fernández-Navarro, and César Hervás-Martínez, *Senior Member, IEEE*

Abstract—Ordinal regression problems are those machine learning problems where the objective is to classify patterns using a categorical scale which shows a natural order between the labels. Many real-world applications present this labelling structure and that has increased the number of methods and algorithms developed over the last years in this field. Although ordinal regression can be faced using standard nominal classification techniques, there are several algorithms which can specifically benefit from the ordering information. Therefore, this paper is aimed at reviewing the state of the art on these techniques and proposing a taxonomy based on how the models are constructed to take the order into account. Furthermore, a thorough experimental study is proposed to check if the use of the order information improves the performance of the models obtained, considering the most significant published approaches within the taxonomy. The results confirm that ordering information benefits ordinal models improving their accuracy and the closeness of the predictions to actual targets in the ordinal scale.

Index Terms—Ordinal regression, ordinal classification, binary decomposition, threshold methods, augmented binary classification, proportional odds model, support vector machines, discriminant learning, artificial neural networks

1 Introduction

Learning to classify or to predict numerical values from prelabelled patterns is one of the central research topics in machine learning and data mining [1]–[3]. However, less attention has been paid to ordinal regression (also called ordinal classification) problems, where the labels of the target variable exhibit a natural ordering. For example, student satisfaction surveys usually involve rating teachers based on an ordinal scale $\{poor, average, good, very\ \ good, excellent\}$. Hence, class labels are imbued with order information, e.g. a sample vector associated with class label $average$ has a higher rating (or better) than another from the $poor$ class, but $good$ class is better than both. When dealing with this kind of problems, two facts are decisive: misclassification costs are not the same for different errors (it is clear that misclassifying an $excellent$ teacher as $poor$ should be more penalised than misclassifying him/her as $very\ good$) and the ordering information can be used to construct more accurate models. A further distinction is made by Anderson [4], which differentiates two major types of ordinal categorical variables, "grouped continuous variables" and "assessed ordered categorical variables". The first one is a discretised version of an underlying continuous variable, which could be observed itself. The second one covers those variables where a user provides his/her judgement on the grade of the ordered categorical variable. However, imposing an ordering is meaningful for both cases.

Ordinal regression problems are very common in many research areas, and they have been frequently considered as standard nominal problems which can lead to non-optimal solutions. Indeed, ordinal regression problems can be said to be between classification and regression, presenting some similarities and differences. Some of the fields where ordinal regression is found are medical research [5]–[11], age estimation [12], brain computer interface [13], credit rating [14]–[17], econometric modelling [18], face recognition [19]–[21], facial beauty assessment [22], image classification [23], wind speed prediction [24], social sciences [25], text classification [26], and more. All these works are examples of application of specifically designed ordinal regression models, where the ordering consideration improves their performance with respect to their nominal counterparts.

In statistics, ordinal data were firstly studied by using a link function able to model the underlying probability for generating ordinal labels [4]. The field of ordinal regression has evolved in the last decade, with a plethora of noteworthy research progress made in supervised learning [27], from support vector machine (SVM) formulations [28], [29] to Gaussian processes [30] or discriminant learning [31], to name a few. However, up to the authors' knowledge, these methods have not yet been categorised in a general taxonomy, which is essential for further research and for identifying the developments made and the present state of existing methods. This paper contributes a review of the state-of-the-art of ordinal regression, a taxonomy proposal to

This work has been partially subsidised by the TIN2011-22794 project of the Spanish Ministry of Economy and Competitiveness (MINECO), FEDER funds and the P2011-TIC-7508 project of the "Junta de Andalucía" (Spain).
P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero and C. Hervás-Martínez are with the Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein building, 14017 - Córdoba, Spain, e-mail:{pagutierrez,i82perom,jsanchezm,chervas}@uco.es
F. Fernández-Navarro is with the Department of Mathematics and Engineering, Loyola University Andalucía, Spain, e-mail: i22fenaf@uco.es

better organise the advances in this field, and an experimental study with a complete repository of datasets and a total of 16 ordinal regression methods (including a software tool to run and test all the methods).

Several objectives motivate the experimental study. First of all, our focus is on evaluating the necessity of taking ordering information into account. In [32], ordinal meta-models were compared with respect to their nominal counterparts to check their ability to exploit ordinal information. The work concludes that such meta-methods do exploit ordinal information and may yield better performance. However, as will be analysed in this work, specifically designed ordinal regression methods can further improve the results with respect to meta-model approaches. Another study [33] argues that ordinal classifiers may not present meaningful advantages over the analogue non-ordinal methods, based on accuracy and Cohen's Kappa statistic [34]. The results of the present review show that statistically significant differences are found when using measures which take the order into account, which is the case of the Mean Absolute Error (MAE), i.e. the average deviation between predicted and actual targets in number of categories. The second main motivation of this paper is to provide some guidelines to decide on the best methods in terms of accuracy, MAE and computational time. Since there are not specific repositories of ordinal regression datasets, proposals are usually evaluated using discretised regression ones, where the target variable is simply divided into different bins or classes. 24 of these discretised datasets are used for our study, in addition to 17 real benchmark ordinal regression datasets extracted from public repositories. The last objective is to evaluate whether the methods behave differently depending on the nature of the datasets.

This paper is a significant extension of a preliminary conference version [35]: a deeper analysis of the state-of-the-art has been performed, including most recent proposals and a taxonomy to group them. Moreover, the experimental study includes more methods and datasets. The rest of the paper is organised as follows. Section 2 introduces the problem of ordinal regression and briefly describes its differences from some related machine learning topics outside the scope of this paper. Section 3 revises ordinal regression state-of-the-art by grouping different methods with a proposed taxonomy. The main representatives of each family are then empirically compared in Section 4, where the experiments are described and the corresponding results are studied and discussed. Finally, Section 5 deals with the main achievements.

2 Notation and Nature of the Problem

2.1 Problem definition

The ordinal regression problem consists on predicting the label y of an input vector $\mathbf{x}$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ and $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$, i.e. $\mathbf{x}$ is in a K-dimensional input space and y is in a label space of Q different labels corresponding to the categories. The objective is to find a classification rule or function $f : \mathcal{X} \to \mathcal{Y}$ to predict the labels of new patterns, given a training set of N points, $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$. A natural label ordering is included for ordinal regression, $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \ldots \prec \mathcal{C}_Q$, where $\prec$ is an order relation given by the nature of the classification problem. Many ordinal regression measures and algorithms consider the rank of the label, i.e. the position of the label in the ordinal scale, which can be expressed by the function $\mathcal{O}(\cdot)$, in such a way that $\mathcal{O}(\mathcal{C}_q) = q, q = 1, \ldots, Q$. The difference between this setting and other related ones is now established. The assumption of an order between class labels makes that two different elements of $\mathcal{Y}$ can be always compared by using the relation $\prec$, which is not possible under the nominal classification setting. If compared to regression (where $y \in \mathbb{R}$), it is true that real values in $\mathbb{R}$ can be ordered by the standard $<$ operator, but labels in ordinal regression ($y \in \mathcal{Y}$) do not carry metric information, so the category serves as a qualitative indication of the pattern rather than a quantitative one.

2.2 Ordinal regression in the context of ranking and sorting

Although ordinal regression has been paid attention recently, the amount of related research topics is worth to be mentioned. First, it is important to remark the differences between ordinal regression and other related ranking problems. There are three terms to be clarified: *ranking*, *sorting* and *multipartite ranking*.

Ranking generally refers to those problems where the algorithm is given a set of ordered labels [36], with one label for each pattern, and the objective is to learn a rule able to rank patterns by using this discrete set of labels. The induced ordering should be partial with respect to the patterns, in the sense that ties are allowed. This rule should be able to obtain a good ranking, but not to classify patterns in the correct class. For example, if the labels predicted by a classifier are shifted one category (in the ordinal scale) with respect to the actual ones, the classifier will still be a perfect ranker.

Another term, *sorting* [36] is referred to the problem where the algorithm is given a total order for the training dataset and the objective is to rank new sets during the test phase. As we can see, this is equivalent to a ranking problem where the size of the label set is equal to the number of training points, $Q = N$. Ties are not allowed for the prediction. Again, the interest is in learning a function that can give a total ordering of the patterns instead of a concrete label.

The *multipartite ranking* problem is a generalisation of the well-known bipartite ranking one. ROC analysis, which evaluates the ability of classifiers to sort positive and negative instances in terms of the area under the ROC curve, is a clear example of training a binary classifier to perform well in a bipartite ranking problem. Multipartite ranking can be seen as an intermediate point

between ranking and sorting. It is similar to ranking because training patterns are labelled with one of Q ordered ratings ($Q = 2$ for bipartite ranking), but here the goal is to learn from them a ranking function able to induce a total order in accordance with the given training ratings [37]–[39], which is similar to sorting. The objective of multipartite ranking is to obtain a classifier which ranks "high" classes ahead of "low" classes (in the ordinal scale), being this a refinement of the order information provided by an ordinal classifier, as the latter does not distinguish between objects within the same category. The relationship between multipartite ranking and ordinal classification is discussed in [38]. An ordinal regression classifier can be used as a ranking function by interpreting the class labels as scores. However, this type of scoring will produce a large number of ties (which is not desirable for multipartite ranking). On the other hand, a multipartite ranking function $f(\cdot)$ can be turned into an ordinal classifier by deriving thresholds to define an interval for each class, but how to find the optimal thresholds is an open issue.

A more general term is *learning to rank*, gathering different methods in which the goal is to automatically construct a ranking model from training data [40]. Methods used for the three previously mentioned problems can be used for *learning to rank* ones. Moreover, ordinal regression can be used as a *learning to rank* algorithm, where the categories are individually evaluated for each training pattern, using a finite ordinal scale. In this context, we refer now to the categorisation presented in [40], which establishes different families of ranking model structures: *pointwise* or *itemwise ranking* (where the relevance of an input vector $\mathbf{x}$ is predicted by using either real-valued scores or ordinal labels), *pairwise ranking* (where the relative order between two input vectors $\mathbf{x}$ and $\mathbf{x}'$ is tried to be predicted, i.e. the local comparison nature of ranking, which can be easily cast to binary classification) and *listwise ranking* (where the algorithms try to order a finite set of patterns $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ by minimising the inconsistency between the predicted permutation and the training permutation. All these categories are connected to ordinal regression in [41].

In summary, ordinal regression is a pointwise approach to classify data, where the labels exhibit a natural order. It is related to the problems of ranking, sorting and multipartite ranking, but, during the test phase, its objective is to obtain correct labels or labels as close as possible to the correct ones, not a correct relative partial order of the patterns (ranking), a total order of patterns in accordance to the order of the training set (sorting) or a total order in accordance to the training labels (multipartite ranking).

2.3 Advanced related topics

In this section, other advanced methods related to ordinal regression are surveyed. They are outside the scope of this paper, as they consider different learning settings

Monotonic classification [42]–[44] is a special class of ordinal classification task, where there are monotonicity constraints between features and decision classes, i.e. $\mathbf{x} \succeq \mathbf{x}' \rightarrow f(\mathbf{x}) \geq f(\mathbf{x}')$ [45], where $\mathbf{x} \succeq \mathbf{x}'$ means that $\mathbf{x}$ dominates $\mathbf{x}'$, i.e. $x_k \geq x'_k, k = 1, \ldots, K$. Monotonic classification tasks are very common in real-world problems [43] (e.g. consider the case where employers must select their employees based on their education and experience), where monotonicity may be an important model requirement for justifying the decision made. This kind of problems have been approached, for example, by decision trees [43], [46] and rough set theory [44].

A recent work is concerned with transductive ordinal regression [27], where a SVM model is derived to learn from a set of labelled and unlabelled patterns. The core of their formulation is an objective function that caters to several commonly used loss functions in transductive settings, but for ordinal regression. This SVM model is combined with a proposed label swapping scheme for multiple class transduction to derive ordinal decision boundaries that pass through a low-density region of the augmented labelled and unlabelled data. Another related work [47] considers transfer learning in the same context, where the objective is to obtain a classifier for new target domains using the available label information of other related source domains. The proposed method spans the feasible solution space with an ensemble of ordinal classifiers from the multiple relevant source domains, using the maximum margin criterion.

Uncertainty has been included in ordinal regression models in two different ways. Nondeterministic ordinal classifiers (defined as those allowed to predict more than one label for some patterns) are considered in [48]. In [49] a kernel model is proposed for those ordinal problems where partial class memberships probabilities are available instead of crisp labels.

One step forward [50] considers those problems where the prediction labels follow a circular order (e.g. directional predictions).

3 An Ordinal Regression Taxonomy

In this section, a taxonomy of ordinal regression methods is proposed. With this purpose we firstly review what have been referred to as *naïve approaches*, in the sense that the model is obtained by using other standard machine learning prediction algorithms (e.g. nominal classification or standard regression). Secondly, *ordinal binary decomposition approaches* are reviewed, the main idea being to decompose the ordinal problem into several binary ones, which are separately solved by multiple models or by one multiple-output model. The third group will include the set of methods known as *threshold models*, which are based on the general idea of approximating a real value predictor and then dividing the real line into intervals. The taxonomy proposed is given in Fig. 1.

3.1 Naïve approaches

Ordinal regression problems can be easily simplified into other standard problems, which generally involves making some assumptions. As will be later discussed, these methods can be very competitive given that, even though these assumptions may not hold, they inherit the performance of very well-tuned models.

3.1.1 Regression

One idea is to cast all the different labels $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ into real values $\{r_1, r_2, \ldots, r_Q\}$ [51], where $r_i \in \mathbb{R}$, and then to apply standard regression techniques [2], [52], [53] (such as neural networks, support vector regression...). Typically, the value of each label is related to its position in the ordinal scale, i.e. $r_i = i$. For example, Kramer et al. [54] map the ordinal scale by assigning numerical values, applying a regression tree model and rounding the results for assigning the class when predicting new values. They also evaluate the possibility of using the median, the mode, or the rounded mean of all the patterns in the leaves of the tree. The main problem with these approaches is that real values used for the labels may hinder the performance of the regression algorithms, and there is no principled way of deciding the value a label should have without prior information about the problem, since the distance between classes is unknown. Moreover, regression learners will be more sensitive to the representation of the label rather than its ordering [55]. A recent alternative is proposed in [56], where, instead of choosing arbitrary ordered values for the different labels, the variable is reconstructed by examining the different pairwise class distances.

3.1.2 Nominal classification

Ordinal classification problems are usually considered from a standard nominal perspective, and the order between classes is simply ignored. Some researchers routinely apply nominal response data analysis methods (yielding results invariant to the permutation of the categories) to both nominal and ordinal target variables alike because they are both categorical [57]. Nominal classification algorithms ignore the ordering of the labels, thus requiring more training data [55]. The Support Vector Machine paradigm (SVM) [58] is perhaps the most common kernel learning method for statistical pattern recognition. Beyond the application of the kernel trick to allow non-linear decision discriminants, and the slack-variables to avoid inseparability, relax the constraints and handle noisy data, the original binary SVM had to be reformulated to deal with multiclass problems [59].

3.1.3 Cost-sensitive classification

A more advanced method that can be considered in this group is cost-sensitive learning. Many real-world applications of machine learning and data mining require the evaluation of the learned system with different costs for different types of misclassification errors [60]. This is the case with ordinal regression, although the exact costs for misclassification can not be always evaluated a priori. The cost of misclassifications can be forced to be different depending on the distance between real and predicted classes, in the ordinal scale. The work of Kotsiantis and Pintelas [61] considers cost-sensitive classification, by using absolute costs (i.e. the element c_{ij} of the cost matrix $\mathbf{C}$ is equal to the difference in the number of categories, $c_{ij} = |i - j|$). Different algorithms are shown to obtain better MAE values when cost matrices are used, without harming (in fact even improving) accuracy [61]. We include two cost matrices for a five class problem in Table 1, with the absolute cost matrix and the quadratic cost ($c_{ij} = |i - j|^2$), together with a zero-one cost matrix, which is the one assumed in nominal classification. Other possibilities are to choose asymmetric costs or non-convex two-Gaussian cost [41]. Again, the main problem is that, without a priori knowledge of the ordinal regression problem, it is not clear which cost matrix is more suitable.

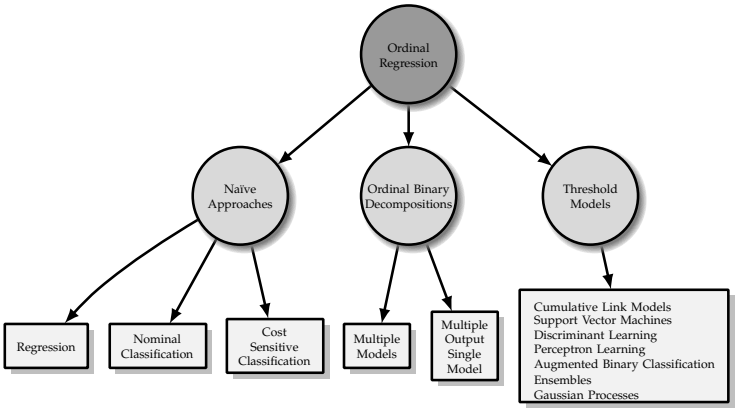


Fig. 1. Proposed taxonomy for ordinal regression methods

TABLE 1
Example of different cost matrices for a five class classification problems, with class labels $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$.

Zero-one	Absolute cost	Quadratic cost
$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 4 & 9 & 16 \\ 1 & 0 & 1 & 4 & 9 \\ 4 & 1 & 0 & 1 & 4 \\ 9 & 4 & 1 & 0 & 1 \\ 16 & 9 & 4 & 1 & 0 \end{pmatrix}$

Actual class labels are arranged in rows, while predicted class labels are arranged in columns.

3.2 Ordinal binary decompositions

This group includes all those methods which are based on decomposing the ordinal target variable into several binary ones, which are then estimated by a single or multiple models. A summary of the decompositions is given in Table 2, where five classes are considered, each

TABLE 2
Binary decompositions for a 5-class ordinal problem, with class labels $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$.

Nominal decompositions	
OneVsAll	*OneVsOne*
$\begin{pmatrix} +,-,-,-,- \\ -,+,-,-,- \\ -,-,+,-,- \\ -,-,-,+,- \\ -,-,-,-,+ \end{pmatrix}$	$\begin{pmatrix} -,-,-,-, , , , , \\ +, , , ,-,-,-, , , \\ ,+, , ,+, , ,-,-, \\ , ,+, , , ,+, ,+, ,- \\ , , ,+, , , ,+, ,+ \end{pmatrix}$

Ordinal decompositions			
OrderedPartitions	*OneVsNext*	*OneVsFollowers*	*OneVsPrevious*
$\begin{pmatrix} -,-,-,- \\ +,-,-,- \\ +,+,-,- \\ +,+,+,- \\ +,+,+,+ \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +,-, , \\ ,+,-, \\ , ,+,- \\ , , ,+ \end{pmatrix}$	$\begin{pmatrix} -, , , \\ +,-, , \\ +,+,-, \\ +,+,+,- \\ +,+,+,+ \end{pmatrix}$	$\begin{pmatrix} +,+,+,+ \\ +,+,+,- \\ +,+,-, \\ +,-, , \\ -, , , \end{pmatrix}$

method generating a different decomposition matrix. Columns of the matrix correspond to the binary subproblems and rows to the role of each class for each subproblem. The symbol $+$ is associated to the positive class and the symbol $-$ to the negative one. If the class is not used in the specific binary subproblem, no symbol is included in the corresponding position. *OneVsAll* and *OveVsOne* formulations are nominal classification methods (and should be listed as naïve approaches), but they have been included in this table for comparison purposes. Note the high number of binary decompositions needed by *OneVsOne* (in this case, 10 combinations).

Two main issues have to be taken into account when analysing the methods herein presented: 1) some of them are based on the idea of training a different model for each subproblem (multiple model approaches), while others learn one single model for all the subproblems; 2) apart from defining how to decompose the problem, it is important to define a rule for predicting new patterns, once the decision values are obtained. For the prediction phase, the corresponding binary codes of Table 2 can be considered as part of the error-correcting output codes (ECOC) framework [62], where the predicted class is the one closest to the code formed by all binary responses. Taking the first criterion into account, we have divided ordinal binary decomposition algorithms into *multiple model* and *multiple-output single model* approaches.

3.2.1 Multiple model approaches

Ordinal information gives us the possibility of comparing the different labels. For a given rank q, a direct question can be the following, "is the label of pattern $\mathbf{x}$ greater than q?" [41]. This question is clearly a binary classification problem, so ordinal classification can be solved by considering each binary classification problem independently and combining the binary outputs into a label, which is the approach followed by Frank and Hall in [63] (this decomposition is called *OrderedPartitions* in Table 2). In their work, Frank and Hall considered C4.5 as the binary classifier and the decision of the different binary classifiers were combined by using associated probabilities $p_q = P(y \succ \mathcal{C}_q|\mathbf{x}), q = 1, \ldots, Q-1$:

$$P(y = \mathcal{C}_1|\mathbf{x}) \approx 1 - p_1, P(y = \mathcal{C}_Q|\mathbf{x}) \approx p_{Q-1},$$
$$P(y = \mathcal{C}_q|\mathbf{x}) \approx p_{q-1} - p_q,\ 2 \leq q \leq Q-1.$$

Note that this approach may lead to negative probability estimates [64], given that binary classifiers are independently learned and nothing assures that $p_{q-1} < p_q$. When there is no need for proper probability estimations, prediction can be done by selecting the maximum.

In the work of Waegeman et al. [65], this framework is used but explicit weights over the patterns of each binary system are imposed, in such a way that errors on training objects are penalised proportionally to the absolute difference between their rank and q (the category examined). Additionally, labels for the test set are obtained by combining the estimated outcomes y_q of all the $Q-1$ binary classifiers. The interpretation of these binary outcomes $y_{qi} \in \{+1, -1\}, q = 1, \ldots, Q-1, i = 1, \ldots, N$, intuitively leads to $y_i \succ \mathcal{C}_q$ if $y_{qi} = +1$. In this way, the rank k is assigned to pattern $\mathbf{x}_i$ so that $y_{qi} = -1, \forall q < k$, and $y_{qi} = +1, \forall q \geq k$. As stated by the authors, this strategy can result in ambiguities for some test patterns, and they should be solved by using similar techniques to those considered for nominal classification. A very similar scheme is proposed in [12], where the weights are obtained slightly differently, and different kernels are used for the different binary classification sub-problems.

Other ordinal binary decompositions can be found in the literature. The cascade linear utility model [66] considers $Q-1$ projections, in such a way that projection q separates classes $\mathcal{C}_1 \cup \ldots \cup \mathcal{C}_{Q-q-1}$ from class C_{Q-q}, i.e. one class is eliminated for each projection (this is the *OneVsPrevious* decomposition in Table 2). The predictions are then combined by a union utility function. Finally, binary SVMs were also applied to ordinal regression [15], by making use of the ordinal pairwise partitioning approach [14]. This approach is composed of four different reformulations of the classical *OneVsOne* and *OneVsAll* paradigms. *OneVsNext* considers that each binary classifier q separates class $\mathcal{C}_q$ from class $\mathcal{C}_{q+1}$, and *OneVsFollowers* (which is similar to the *OneVsPrevious* approach in [66] but in the opposite direction) constructs each binary classifier q for the task of separating class $\mathcal{C}_q$ from classes $\mathcal{C}_{q+1} \cup \ldots \cup \mathcal{C}_Q$. The prediction phase is then approached by examining each binary classifier in order, so that, if a model predicts that the pattern is in the class which is isolated (not grouped with other classes), then this is the predicted class. This can be done in a forward manner or in a backward manner [15].

Finally, another possibility [67] is to derive a classifier for each class but separating the labels into groups of three classes (instead of only two) for intermediate subtasks (labels lower than $\mathcal{C}_q$, label $\mathcal{C}_q$, and labels higher than $\mathcal{C}_q$), or two classes for the extreme ones. The objective is to incorporate the order information in the subclassification tasks. Although the decomposition for intermediate classes is not binary but ternary, this

approach has been included in this group because its motivation is similar to all the aforementioned.

3.2.2 Multiple-output single model approaches

Among non-parametric models, one appealing property of neural networks is that they can handle multiple responses in a seamless fashion [68]. Usually, as many output neurons as the number of target variables are included in the output layer and targets are presented to the network in the form of vectors $\mathbf{t}_i, i = 1, \ldots, N$. When applied to nominal classification, the most usual approach is to consider a 1-of-Q coding scheme [53], i.e. $\mathbf{t}_i = \{t_{i1}, \ldots, t_{iQ}\}$, $t_{iq} = 1$ if $\mathbf{x}_i$ corresponds to an example belonging to class $\mathcal{C}_q$, and $t_{iq} = 0$ (or $t_{iq} = -1$), otherwise. In the ordinal regression framework, one can take the ordering information into account to design specific ordinal target coding schemes, which can improve the performance of the methods. Indeed, all the decompositions in Table 2 can be used to train neural networks, by taking each row as the code for the target class, $\mathbf{t}_i$, and a single model will be obtained for all related subproblems (considering that each output neuron is solving each subproblem). This can be done by assigning a value ($+1$, 0 or -1) to each of the different symbols ($+$ or $-$) in Table 2. For sigmoidal output neurons, a 1 is assigned for positive symbols ($+$) and a 0 for negative ones ($-$). For hyperbolic functions, negative symbols are represented with a -1 and positive ones also with a 1. Those decompositions where a class is not involved should be treated as a "does not matter" condition where, whatever the output response, no error signal should be generated [69].

A generalisation of ordinal perceptron learning [70] in neural networks was proposed in [71]. The method is based on two main ideas: 1) the targets are coded using the *OrderedPartitions* approach; and 2) instead of using the softmax function [53] for the output nodes, a standard sigmoid function is imposed, and the category assigned to a pattern is equal to the index previous to that of the first output node whose value is higher than a predefined threshold T, or when no nodes are left. This method ignores inconsistencies (e.g. a sigmoid with value higher than T after the index selected).

Extreme learning machines (ELMs) are single-layer feedforward neural networks, where the hidden layer does not need to be tuned given that corresponding weights are randomly assigned. ELMs have demonstrated good scalability and generalisation performance with a faster learning speed when compared to other models such as SVMs [72]. They have been adapted to ordinal regression [73], and one of the proposed ordinal ELMs also considers *OrderedPartitions* targets. Additionally, multiple models are also trained using the *OneVsOne* and the *OrderedPartitions* approaches. For the prediction phase, the loss-based decoding approach [62] is utilised, i.e. the chosen label is that which minimises the exponential loss, $k = \arg\min_{q=1,\ldots,Q} d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right)$, where $\mathbf{M}_q$ is the code associated to class q (q-th row of the coding matrix), $\mathbf{y}(\mathbf{x})$ is the vector of predictions, and $d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right)$ is the exponential loss function, $d\left(\mathbf{M}_q, \mathbf{y}(\mathbf{x})\right) = \sum_{i=1}^{Q} \exp\left(\mathbf{M}_{qi} \cdot \mathbf{y}_i(\mathbf{x})\right)$. The values of the vector $\mathbf{y}(\mathbf{x})$ are assumed to be in the $[-1, +1]$ range, and those of $\mathbf{M}_q$ in the set $\{-1, 0, +1\}$. The single ELM was found to obtain slightly better generalisation results and also to report the lowest computational time [73]. Other adaption of the ELM is found in [74], where an evolutionary algorithm is applied to optimise the different weights of the model by using a fitness function to impose the ordering restriction in model selection. A different approach is taken in [75], where the ordinal constraints are included into the weights connecting the hidden and output layers.

Costa [69] followed a probabilistic framework to propose another neural network architecture able to exploit the ordinal nature of the data. The proposal is based on the joint prediction of constrained concurrent events, which can be turned into a classification task defined in a suitable space through a "partitive approach". An appropriate entropic loss is derived for $\mathbf{P}(\mathcal{Y})$, i.e. the set of subsets of $\mathcal{Y}$, where $\mathcal{Y}$ is a set of Q elementary events. A probability for each possible subset should be estimated, leading to a total of 2^Q probabilities. However, depending on the classification problem, not all possibilities should be examined. For example, this is simplified for random variables taking values in finite ordered sets (i.e. ordinal regression), as well as in the case of independent boolean random variables (i.e. nominal classification). To adapt neural networks to the ordinal case structure, targets were reformulated following the *OneVsFollowers* approach and the prediction phase was accomplished by considering that, under its constrained entropic loss formulation, the output of the q-th output neuron estimates the probability that q and $q-1$ events are both true. This methodology was further evaluated and compared in other works [64], [76], [77].

Although all these neural network approaches consist of a single model, they are trained independently in the sense that the output of the neurons do not depend on the other outputs (only on common nonlinear transformations of the inputs). That is the reason why we have included them into the category of ordinal binary decompositions.

These neural network models can be grouped under the term multitask learning [78] (MTL), which is a learning paradigm that considers the case of simultaneously tackling several related tasks. Any of the different proposals in this field could be applied to train a single model for the different ordinal decompositions analysed in this section. Indeed, one of the existing proposals, MTL via conic programming [79], was validated in the context of ordinal regression, showing promising results.

3.3 Threshold models

Often, in the ordinal regression paradigm, it is natural to assume that an unobserved continuous variable underlies the ordinal response variable. Such a variable

is called a latent variable, and methods based on that assumption are known as threshold models, which are the most popular approaches for modelling ordinal and ranking problems [49]. These methodologies estimate:

- A function $f(\mathbf{x})$ that tries to predict the nature of those underlying real-valued outcomes, which acts as a projection function (similar to the ranking function to be learned by multipartite algorithms).
- A set of thresholds $\mathbf{b} = (b_1, b_2, \dots, b_{Q-1}) \in \mathbb{R}^{Q-1}$ to represent intervals in the range of $f(\mathbf{x})$, which must satisfy the constraints $b_1 \leq b_2 \leq \dots \leq b_{Q-1}$.

Threshold models can be seen as an extension of naïve regression models. The main difference between these two approaches is that the distances among the different classes are not defined a priori for threshold models, being estimated during the learning process. Although they are also related to (single-model) ordinal binary decomposition approaches, inconsistencies in the predictions can be found for the latter kind of models.

3.3.1 Cumulative link models

Arising from a statistical background, the Proportional Odds Model (POM) is one of the first models specifically designed for ordinal regression [80], dated back to 1980. It is a member of a wider family of models recognised as Cumulative Link Models (CLMs) [81]. In order to extend binary logistic regression to ordinal regression, CLMs predict probabilities of groups of contiguous categories, taking the ordinal scale into account. In this way, cumulative probabilities $P(y \preceq \mathcal{C}_j|\mathbf{x})$ are estimated, which can be directly related to standard probabilities:

$$P(y \preceq \mathcal{C}_q|\mathbf{x}) = P(y = \mathcal{C}_1|\mathbf{x}) + \dots + P(y = \mathcal{C}_q|\mathbf{x}),$$
$$P(y = \mathcal{C}_q|\mathbf{x}) = P(y \preceq \mathcal{C}_q|\mathbf{x}) - P(y \preceq \mathcal{C}_{q-1}|\mathbf{x}),$$

with $q = 1, \dots, Q$, and considering by definition that $P(y \preceq \mathcal{C}_Q|\mathbf{x}) = 1$. Stochastic ordering of space $\mathcal{X}$ is satisfied by the following general model form [28]:

$$g^{-1}\left(P(y \preceq \mathcal{C}_q|\mathbf{x})\right) = b_q - \mathbf{w}^T\mathbf{x}, q = 1, \dots, Q,$$

where $g^{-1} : [0, 1] \to (-\infty, +\infty)$ is a monotonic function often referred to as the inverse link function and b_q is the threshold defined for class $\mathcal{C}_q$. This model is clearly inspired by the latent variable motivation, considering that $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is a linear transformation. A decision rule $f : \mathcal{X} \to \mathcal{Y}$ is not fitted directly. If the ordinal response is a coarsely measured latent continuous variable $f(\mathbf{x})$, label $\mathcal{C}_q$ in the training set is observed if and only if $f(\mathbf{x}) \in [b_{q-1}, b_q]$, where the function f (latent utility) and $\mathbf{b} = (b_0, b_1, ..., b_{Q-1}, b_Q)$ are to be determined from the data. It is assumed that $b_0 = -\infty$ and $b_Q = +\infty$, so the real line, defined by $f(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$, is divided into Q consecutive intervals. Each region separated by two consecutive biases corresponds to a category $\mathcal{C}_q$. The constraints $b_1 \leq b_2 \leq \dots \leq b_{Q-1}$ ensure that $P(y \preceq \mathcal{C}_q|\mathbf{x})$ increases with q [82].

Suppose a model of the latent variable, $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+\epsilon$, where ϵ is the random component with zero expectation, $\mathbf{E}[\epsilon] = 0$, distributed according to F_ϵ. If a distribution assumption F_ϵ is made for ϵ, the cumulative model is obtained by choosing the inverse distribution F_ϵ^{-1} as the inverse link function g^{-1}. The most common choice for the distribution of ϵ is the logistic function (which is indeed the one selected for the POM [83]), although probit, complementary log-log, negative log-log or cauchit functions could also be used [81]. As will be seen, all the models in this section are inspired by the POM in the strategy assumed, obtaining a projection and dividing this projection into different ordered intervals. This projection can be used to obtain more information about the confidence of the predictions by relating it to its proximity to the biases. Additionally, the POM model provides us with a solid probabilistic interpretation.

Focusing on the POM model [83], the distribution of ϵ is assumed to be the standard logistic function:

$$g^{-1}\left(P(y \preceq \mathcal{C}_q|\mathbf{x})\right) = \ln\left(\frac{P(y \preceq \mathcal{C}_q|\mathbf{x})}{P(y \succ \mathcal{C}_q|\mathbf{x})}\right) = b_q - \mathbf{w}^T\mathbf{x},$$

where $q = 1, \dots, Q-1$, $odds(y \preceq \mathcal{C}_q|\mathbf{x}) = \exp(b_q - \mathbf{w}^T\mathbf{x})$, so $odds(y \preceq \mathcal{C}_q|\mathbf{x}) = \frac{P(y \preceq \mathcal{C}_q|\mathbf{x})}{1-P(y \preceq \mathcal{C}_q|\mathbf{x})}$. Therefore, the ratio of the odds for two pattens $\mathbf{x}_0$ and $\mathbf{x}_1$ are proportional:

$$\frac{odds(y \preceq \mathcal{C}_q|\mathbf{x}_1)}{odds(y \preceq \mathcal{C}_q|\mathbf{x}_0)} = \exp(-\mathbf{w}^T(\mathbf{x}_1 - \mathbf{x}_0)).$$

More flexible non-proportional alternatives have been developed, one of them simply assuming different $\mathbf{w}$ for each class (which is known as the generalised ordered logit model [84]), and another applying the proportional odds assumption only to a subset of variables (partial proportional odds [85]). Moreover, Tutz [86] presented a general framework for parametric models that extends generalised additive models to incorporate nonparametric parts.

Another main problem with linear CLMs is that they are rather inflexible since the decision functions are always linear hyperplanes, this generally affecting the performance of the model (as analysed in the experimental section of this work). A non-linear version of the POM model was proposed in [18], [82] by simply setting the projection $f(\mathbf{x})$ to be the output of a neural network. The probabilistic interpretation of CLMs can be used to apply a maximum likelihood maximisation for setting the network parameters. Gradient descent techniques with proper constraints for the biases serve this purpose. This non-linear generalisation of the POM model based on neural networks was considered in [87], where an evolutionary algorithm was applied to optimise all the parameters considered. Linear ordinal logistic regression was combined with nonlinear kernel machines using primal-dual relations from Nystrom sampling [88]. However, to make the computation of the model feasible, a sub-sample from the data had to be selected, which limits the applicability to those cases where there is a reasonable way to do this [88].

An interesting alternative to CLMs is the so-called ordistic model presented in [89]. The work presents

two threshold-based constructions which can be used to generalise loss functions for binary labels, such as the logistic and hinge loss, and another generalisation of the logistic loss based on a probabilistic model for ordered labels. Both constructions are based on including $Q-1$ thresholds partitioning the real line to Q segments, but they differ in how predictors outside the "correct" segment (or too close to its edges) are penalised. The immediate-threshold construction only penalises the violations of the two thresholds limiting this segment, while the all-threshold one considers all of them.

3.3.2 Support vector machines

Because of their good generalisation performance, SVM models are maybe the most widely applied ones to ordinal regression, their structure being easily adapted to that of threshold models. The proposal of Herbrich et al. [28], [90] is the first SVM based algorithm, where they consider a pairwise approach by deriving a new dataset made up of all possible difference vectors $\mathbf{x}_{ij}^d = \mathbf{x}_i - \mathbf{x}_j$ and $y_{ij} = \text{sign}\left(\mathcal{O}(y_i) - \mathcal{O}(y_j)\right)$, with $y_i, y_j \in \{\mathcal{C}_1, \ldots, \mathcal{C}_Q\}$. In contrast, all the SVM pointwise approaches share the common objective of seeking $Q-1$ parallel discriminant hyperplanes, all of them represented by a common vector $\mathbf{w}$ and the scalars biases $b_1 \leq \ldots \leq b_{Q-1}$ to properly separate training data into ordered classes. In this sense, several methodologies for the computation of $\mathbf{w}$ and $\{b_1, \ldots, b_{Q-1}\}$ can be considered. The work of Shashua and Levin [91] introduced two first methods: the maximisation of the margin between the closest neighbouring classes and the maximisation of the sum of margins between classes. Both approaches present two main problems [64]: the model is incompletely specified, because the thresholds are not uniquely defined, and they may not be properly ordered at the optimal solution, since the inequality $b_1 \leq b_2 \leq \ldots \leq b_{Q-1}$ is not included in the formulation.

Consequently, Chu and Keerthi [29], [92] proposed two different reformulations for the same idea, solving the problem of unordered thresholds at the solution. On the one hand, they imposed explicit constraints on the optimisation problem, only considering adjacent labels for threshold determination (Support Vector Ordinal Regression with Explicit Constraints, SVOREX). On the other hand, patterns in all the categories were allowed to contribute errors for each hyperplane (SVOR with Implicit Constraints, SVORIM), which, as they prove [29], leads to automatically satisfied constraints in the optimal solution. More specifically, the SVORIM learning problem is defined as follows:

$$\min_{\mathbf{w},\mathbf{b},\xi,\xi^*} \frac{1}{2}||\mathbf{w}|| + C\sum_{q=1}^{Q-1}\left(\sum_{j=1}^{q}\sum_{i=1}^{N_q}\xi_{ji}^q + \sum_{j=q+1}^{Q}\sum_{i=1}^{N_q}\xi_{ji}^{*q}\right),$$

subject to the constraints: $\mathbf{w}\cdot\mathbf{x}_i^j - b_k \leq -1 + \xi_{ji}^q,\ \ \xi_{ji}^q \geq 0$ (for $j \in \{1,\ldots,q\},\ i \in \{1,\ldots,N_q\}$), and $\mathbf{w}\cdot\mathbf{x}_i^j - b_{k+1} \geq +1 - \xi_{ji}^{*q},\ \ \xi_{ji}^{*q} \geq 0$ (for $j \in \{q+1,\ldots,Q\},\ i \in \{1,\ldots,N_q\}$), where $\mathbf{b} \in \mathbb{R}^{Q-1}$, ξ_{ji}^q and ξ_{ji}^{*q} are the slacks for the q-th parallel hyperplane (defined for the left and right part of the hyperplanes, respectively), and N_q is the number of patterns of class $\mathcal{C}_q$. They empirically found that SVOREX performed better in terms of accuracy (with a more local behaviour), and SVORIM preceded in terms of absolute deviations in number of classes or MAE (with a more global behaviour), and this is justified theoretically based on the loss minimised for each method. The framework of reduction [41] also explains this from the point of view of the cost matrices selected. Our results seem to agree with these conclusions for discretised regression datasets, but the differences are not so clear for real ordinal regression ones. Generalisation properties for some ordinal regression algorithms, including SVOR, were further studied in [93].

In [94], the errors of an ordinal SVM classifier are studied separately depending on whether they correspond to upgrading errors (predicted label higher than the actual one) or downgrading ones (the predicted label being lower than the actual one). Authors address the two-objective problem of finding a classifier maximising simultaneously the two margins, and they show that the whole set of Pareto-optimal solutions can be obtained by solving a quadratic optimisation problem.

Some recent works focused on solving the bottleneck of these SVM proposals, which is usually the high computational complexity to handle larger datasets. Concerning this topic, two different proposals can be distinguished: block-quantised support vector ordinal regression [95] and ordinal-class core vector machines [96]. The former is based on performing kernel k-means and applying SVOR in the cluster representatives, on the idea of approximating the kernel matrix K by $\tilde{K}$ which will be composed of k^2 constant blocks, in such a way that the problem scales with the number of clusters, instead of the dataset size. The latter is an extension of core vector machines [97] in the ordinal regression setting. Finally, an incremental version of SVOR algorithms is proposed in [98].

3.3.3 Discriminant learning

Discriminant learning has also been reformulated to tackle ordinal regression [31]. Discriminant analysis is usually not considered as a classification technique by itself, but rather as a supervised dimensionality reduction. Nonetheless, it is widely used for that purpose, since, as a projection method, the definition of thresholds can be used to discriminate the classes. In general, to allow the computation of the optimal projection for the data, this algorithm analyses two main objectives: the maximisation of the between-class distance, and the minimisation of the within-class distance, by using variance-covariance matrices and the Rayleigh coefficient. In order to reformulate the algorithm for ordinal regression, an ordering constraint over contiguous classes is imposed on the averages of projected patterns of each class, which leads the algorithm to order projected patterns according

to their label. This will preserve the ordinal information and avoid some serious ordinal misclassification errors. The original optimisation problem is transformed and extended with a penalty term (C):

$$\min J(\mathbf{w}, \rho) = \mathbf{w}^{\mathrm{T}} S_w \mathbf{w} - C\rho,$$

subject to $\mathbf{w}^{\mathrm{T}} (\mathbf{m}_{q+1} - \mathbf{m}_q) \geq \rho$, where $\mathbf{m}_q = \frac{1}{N_q} \sum_{i=1}^{N_q} \mathbf{x}_i$, S_w is the the between-class scatter matrix and ρ represents the minimum difference of the projected means between consecutive classes (if $\rho > 0$, the projected means are correctly ranked). This methodology is known as Kernel Discriminant Learning for Ordinal Regression (KDLOR) [31] and it has been used in some later works [9], [99]. In [100], the KDLOR model is extended by trying to learn multiple orthogonal projections, which are then combined into a final decision function.

The method was extended in [101], [102] based on the idea of preserving the intrinsic geometry of the data in the embedded non-linear structure, i.e. in the induced high-dimensional feature space, via kernel mapping. This consideration is the basis of manifold learning [53], and the algorithms mentioned construct a neighbourhood graph (which takes the ordinal nature of the dataset into account) which is afterwards used to derive the Laplacian matrix and obtain a projection which considers the underlying manifold of the data. A related method is proposed in [103], where several different projections are iteratively derived.

3.3.4 Perceptron learning

PRank [104] is a perceptron online learning algorithm with the structure of threshold models. It was then extended by approximating the Bayes point, what provides good performance for generalisation [55].

3.3.5 Augmented binary classification

Although the approaches in Subsection 3.2 are simple to implement, their generalisation performance cannot be analysed easily. The two algorithms included in this subsection work differently, and, as later analysed, the models derived are equivalent to threshold models.

A reduction framework can be found in the works of Lin and Li [41], [105], where ordinal regression is reduced to binary classification, applying three steps:

1) Given a coding matrix $\mathbf{M}$ of $(Q-1)$ rows, input patterns $(\mathbf{x}_i, y_i)$ are transformed into extended binary patterns by replicating them, $(\mathbf{x}_i^{(q)}, y_i^{(q)})$, with:

$$\mathbf{x}_i^{(q)} = (\mathbf{x}_i, \mathbf{m}_q),\ y_i^{(q)} = 2[\![q < \mathcal{O}(y_i)]\!] - 1,$$

where $q = 1, \ldots, Q-1$, $\mathbf{m}_q$ is the q-th row of $\mathbf{M}$ and $[\![\cdot]\!]$ is a Boolean test which is 1 if the inner condition is true, and 0 otherwise. $Q-1$ replicates of each pattern are generated with weights:

$$w_{i,q} = (Q-1) \cdot |C_{\mathcal{O}(y_i),q} - C_{\mathcal{O}(y_i),q+1}|,$$

where $i = 1, \ldots, N$, C is a V-shaped cost matrix (i.e. $C_{\mathcal{O}(y_i),q-1} \geq C_{\mathcal{O}(y_i),q}$, if $q \leq \mathcal{O}(y_i)$, and $C_{\mathcal{O}(y_i),q} \leq C_{\mathcal{O}(y_i),q+1}$, if $q \geq \mathcal{O}(y_i)$). The cost matrix must be defined a priori. An example of this transformation is given in Table 3.

2) A single binary classifier with confidence outputs, $f(\mathbf{x}, \mathbf{m}_k)$, is trained for the new extended patterns, aiming at a low weighted $0/1$ loss.
3) A classification rule like the following is used to construct a final prediction for new patterns:

$$r(\mathbf{x}) = 1 + \sum_{q=1}^{Q-1} [\![f(\mathbf{x}, \mathbf{m}_q) > 0]\!]. \quad (1)$$

TABLE 3
Extended binary transformation for three given patterns $(\mathbf{x}_1, y_1 = \mathcal{C}_1), (\mathbf{x}_2, y_2 = \mathcal{C}_2), (\mathbf{x}_3, y_3 = \mathcal{C}_3)$, the identity coding matrix and the quadratic cost matrix.

			$\mathbf{x}_i^{(q)}$		
i	q	$w_{i,q}$	$\mathbf{x}$	$\mathbf{m}_q$	$y_i^{(q)}$
1	1	$2 \cdot \lvert 0-1 \rvert = 2$	$\mathbf{x}_1$	$\{1,0\}$	$2[\![1<1]\!]-1=-1$
1	2	$2 \cdot \lvert 1-4 \rvert = 6$	$\mathbf{x}_1$	$\{0,1\}$	$2[\![2<1]\!]-1=-1$
2	1	$2 \cdot \lvert 1-0 \rvert = 2$	$\mathbf{x}_2$	$\{1,0\}$	$2[\![1<2]\!]-1=+1$
2	2	$2 \cdot \lvert 0-1 \rvert = 2$	$\mathbf{x}_2$	$\{0,1\}$	$2[\![2<2]\!]-1=-1$
3	1	$2 \cdot \lvert 4-1 \rvert = 6$	$\mathbf{x}_3$	$\{1,0\}$	$2[\![1<3]\!]-1=+1$
3	2	$2 \cdot \lvert 1-0 \rvert = 2$	$\mathbf{x}_3$	$\{0,1\}$	$2[\![2<3]\!]-1=+1$

All the binary classification problems are solved jointly by computing a single binary classifier. The most striking characteristic of this algorithm is that it unifies many existing ordinal regression algorithms [41], such as the perceptron ones [104], kernel ranking [36], AdaBoost.OR [106], ORBoost-LR and ORBoost-All thresholded ensemble models [107], CLM [81] or several ordinal SVM proposals (oSVM [64], SVORIM and SVOREX [29]). Moreover, it is important to highlight the theoretical guarantees provided by the framework, including the derived cost and regret bounds and the proof of equivalence between ordinal regression and binary classification. An extension of this reduction framework was proposed in [108], where ordinal regression is proved to be equivalent to a regular multiclass classification whose distribution is changed. This extension is free of the following restrictions: target functions should be rank-monotonic; and rows of loss matrix are convex.

The data replication method of Cardoso et al. [64] (whose previous linear version appeared in [10]) is a very similar framework, except that it essentially considers the absolute cost, consequently being less flexible. However, for ordinal regression, increasing the error with the absolute difference between the predicted and estimated labels is a natural choice in the absence of any other information [18]. An advantage of the framework of data replication is that it includes a parameter s which limits the number of adjacent classes considered, in such a way that the replicate q is constructed by using the $q-s$ classes to its 'left' and the $q+s$ classes to its 'right' [64]. This parameter $s \in \{1, ..., Q-1\}$ plays the role of controlling the increase of data points.

It is worth mentioning that augmented binary classification models and threshold models are closely related. The intersection of the binary hyperplane obtained in the extended dataset with each of the subspace replicas can be projected to the original dataset [10] to derive parallel boundaries, resulting in this way in a threshold model. In fact, SVORIM and reduction to SVM is known to be not so different in formulation [103]. The model of Mathieson [18] (threshold model) is equivalent to the one proposed in [64] (oNN, an augmented binary classification model) if the activation function of the output node is set to the *logsig* function and the model is trained to predict the posterior probabilities when fed with the original input variables and the variables generated by the data replication method. The predicted thresholds would be the weights of the connection of the added $Q-2$ components.

3.3.6 Ensembles

From a different perspective, the confidence of a binary classifier can be regarded as an ordering preference. RankBoost [109] is a boosting algorithm that constructs an ensemble of those confidence functions to form a better ordering preference. Some efforts were made to apply a similar idea for ordinal regression problems, deriving into Ordinal Regression Boosting (ORBoost) [107]. The corresponding thresholded-ensemble models inherit the good properties of ensembles, including more stable predictions and sufficient power for approximating complicated target functions [110]. The model is composed of confidence functions, and their weighted linear combination is used as the projection $f(\mathbf{x})$. Following a similar approach to [89], large margin bounds of the classification error and the absolute error are derived, from which two algorithms are presented: ORBoost with all margins and ORBoost with left-right margins [107]. Two alternative thresholded-ensemble algorithms are presented in [111], both generating an ensemble of ordinal decision rules based on forward stagewise additive modelling.

With a different perspective, the well-known AdaBoost algorithm was recently extended to improve any base ordinal regression algorithm [106]. The extension, AdaBoost.OR, proved to inherit the good properties of AdaBoost, improving both the training and test performances of existing ordinal classifiers. Another ordinal regression version of AdaBoost is proposed in [112], while in this case the adaption is based on considering a cost matrix both in pattern weighting and error updating.

The framework of negative correlation learning (where the ensemble members are learnt in such a way that the correlation between their responses is minimised) was used in the context of ordinal regression [17], [113] by calculating the correlation between the latent variable estimations or, alternatively, between the probabilities obtained by the ensemble members.

3.3.7 Gaussian processes

All the previous threshold models can be considered discriminative models in the sense that they estimate directly the posterior $P(y|\mathbf{x})$, or learn a function to map the input $\mathbf{x}$ to class labels. On the contrary, generative models learn a model of the joint probability $P(\mathbf{x}, y)$ of input patterns $\mathbf{x}$ and label y, and make the prediction by a Bayesian framework to estimate $P(y|\mathbf{x})$.

Gaussian Processes for Ordinal Regression (GPOR) [30] models the latent variable $f(\mathbf{x})$ using Gaussian Processes, to estimate then all the parameters by means of a Bayesian framework. The values of the latent function $\{f(\mathbf{x}_i)\}$ are assumed to be the given by random variables indexed by their input vectors in a zero-mean Gaussian process. Mercer kernel functions approximate the covariance between the functions of two input vectors. Given the latent function f, the joint probability of observing the ordinal variables is $P(D|f) = \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))$, and the Bayes theorem is applied to write the posterior probability $P(f|D) = \frac{1}{P(D)} \prod_{i=1}^{N} P(y_i|f(\mathbf{x}_i))P(f)$. A Gaussian noise with zero mean and unknown variance σ^2 is assumed for the latent functions. The normalisation factor $P(D)$, more exactly $P(D|\boldsymbol{\theta})$, is known as the evidence for the vector of hyperparameters $\boldsymbol{\theta}$ and is estimated in the paper by two different approaches: a Maximum a Posteriori approach with Laplace approximation and an Expectation Propagation with variational methods. A more general GPOR was then proposed to tackle multiclass classification problems but with a free structure of preferences over the labels [114]. A probabilistic sparse kernel model was proposed for ordinal regression in [115], where a Bayesian treatment was also employed to train the model. A prior over the weights governed by a set of hyperparameters was imposed, inspired by the well-known relevance vector machine. Srijith et al. have proposed a probabilistic least squares version of GPOR [116], two different sparse versions [117] and a semi-supervised version [118].

3.4 Other approaches and problem formulations

This subsection includes some methods that are difficult to consider in the previous groups. For example, an alternative methodology is proposed by da Costa et al. [76], [77] for training ordinal regression models. The main assumption of their proposal is that the random variable class associated with a given pattern should follow a unimodal distribution. For this purpose, they provide two possible implementations: a parametric one, where a specific discrete distribution is assumed and the associated free parameters are estimated by a neural network; and a non-parametric one, where no distribution is assumed but the error function is modified to avoid errors from distant classes. The same idea was then applied to SVMs in [119] by solving an ordinal problem through a single optimisation process (the all-at-once strategy).

In [120], both decision trees and nearest neighbour (NN) classifiers are applied to ordinal regression problems by introducing the notion of consistency: a small change in the input data should not lead to a 'big jump' in the output decision, i.e. adjacent decision regions should have equal or consecutive labels. This rationale was used as a post-processing mechanism of a standard decision tree and as a pre- or post- processing step for the NN method. An improvement was presented in [121] to reduce the over-regularised decision region artifact by using ensemble learning techniques.

Two ordinal learning vector quantisation schemes, with metric learning, specifically designed for classifying data items into ordered classes, are introduced in [122], [123]. The methods use the order information during training, both in the selection of the prototypes and for determining the way they are updated.

Different prediction methods as a function of the error measure to be minimised are presented in [124]. The paper discusses the fact that the Bayes optimal decision for a classifier which return probability estimates is different depending on the loss function considered for the errors. In this way, for the maximisation of the accuracy one should consider the mode (or maximum probability), but the median of the probability distribution is the optimal decision when minimising the MAE in ordinal regression problems.

4 Experimental Study

This section presents the design of the experimental study followed in this paper (Subsection 4.1), the results obtained for discretised regression datasets (Subsection 4.2) and for ordinal regression ones (Subsection 4.3), and its corresponding discussion (Subsection 4.4).

4.1 Experimental design

In this subsection, the experiments are clearly specified, including the datasets and algorithms considered, the parameters to optimise, the performance measures and the statistical tests used for assessing the differences.

4.1.1 Datasets selected

The most widely used dataset repository is the one provided by Chu et al. [30], including different regression benchmark datasets. These datasets are not real ordinal classification ones but regression problems, which are turned into ordinal classification, the target variable being discretised into Q different bins with equal frequency. It is clear that these datasets do not exhibit some characteristics of typical complex classification tasks, such as class imbalance, given that all classes are assigned the same number of patterns. Furthermore, there are observed values of the actual target regression variable (although they are ignored), so the classification problem can be simpler than problems where these values are not available and there are only categories. On the other hand, we find interesting to check how the algorithms perform in this more controlled environment and to compare the conclusions obtained.

TABLE 4
Characteristics of the benchmark datasets

Discretised regression datasets				
Dataset	#Pat.	#Attr.	#Classes	Class distribution
pyrim5 (P5)	74	27	5	≈ 15 per class
machine5 (M5)	209	7	5	≈ 42 per class
housing5 (H5)	506	14	5	≈ 101 per class
stock5 (S5)	700	9	5	140 per class
abalone5 (A5)	4177	11	5	≈ 836 per class
bank5 (B5)	8192	8	5	≈ 1639 per class
bank5′ (BB5)	8192	32	5	≈ 1639 per class
computer5 (C5)	8192	12	5	≈ 1639 per class
computer5′ (CC5)	8192	21	5	≈ 1639 per class
cal.housing5 (CH5)	20640	8	5	4128 per class
census5 (CE5)	22784	8	5	≈ 4557 per class
census5′ (CEE5)	22784	16	5	≈ 4557 per class
pyrim10 (P10)	74	27	10	≈ 8 per class
machine10 (M10)	209	7	10	≈ 21 per class
housing10 (H10)	506	14	10	≈ 51 per class
stock10 (S10)	700	9	10	70 per class
abalone10 (A10)	4177	11	10	≈ 418 per class
bank10 (B10)	8192	8	10	≈ 820 per class
bank10′ (BB10)	8192	32	10	≈ 820 per class
computer10 (C10)	8192	12	10	≈ 820 per class
computer10′ (CC10)	8192	21	10	≈ 820 per class
cal.housing (CH10)	20640	8	10	2064 per class
census10 (CE10)	22784	8	10	≈ 2279 per class
census10′ (CEE10)	22784	16	10	≈ 2279 per class
Real ordinal regression datasets				
Dataset	#Pat.	#Attr.	#Classes	Class distribution
contact-lenses (CL)	24	6	3	$(15, 5, 4)$
pasture (PA)	36	25	3	$(12, 12, 12)$
squash-stored (SS)	52	51	3	$(23, 21, 8)$
squash-unstored (SU)	52	52	3	$(24, 24, 4)$
tae (TA)	151	54	3	$(49, 50, 52)$
newthyroid (NT)	215	5	3	$(30, 150, 35)$
balance-scale (BS)	625	4	3	$(288, 49, 288)$
SWD (SW)	1000	10	4	$(32, 352, 399, 217)$
car (CA)	1728	21	4	$(1210, 384, 69, 65)$
bondrate (BO)	57	37	5	$(6, 33, 12, 5, 1)$
toy (TO)	300	2	5	$(35, 87, 79, 68, 31)$
eucalyptus (EU)	736	91	5	$(180, 107, 130, 214, 105)$
LEV (LE)	1000	4	5	$(93, 280, 403, 197, 27)$
automobile (AU)	205	71	6	$(3, 22, 67, 54, 32, 27)$
winequality-red (WR)	1599	11	6	$(10, 53, 681, 638, 199, 18)$
ESL (ES)	488	4	9	$(2, 12, 38, 100, 116, 135, 62, 19, 4)$
ERA (ER)	1000	4	9	$(92, 142, 181, 172, 158, 118, 88, 31, 18)$

Table 4 shows the characteristics of the 41 datasets, including the number of patterns, attributes and classes, and also the number of patterns per class. The real ordinal classification datasets were extracted from benchmark repositories[1] (UCI [125] and `mldata.org` [126]), and the regression ones were obtained from the website of W. Chu[2]. For the discretised datasets, we considered $Q = 5$ and $Q = 10$ bins to evaluate the response of the classifiers to the increase in the complexity of the problem. The synthetic `toy` dataset was generated as proposed in [77] with 300 patterns. All nominal attributes were transformed into as many binary attributes as the number of categories and all the datasets were property standardised.

1. We would like to note that many of these datasets have been frequently considered in machine learning literature, ignoring the ordering information.

2. http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html

4.1.2 Algorithms selected

We have selected some representatives of the different families included in the proposed taxonomy (see Table 5). It is important to note that naïve approaches and ordinal binary decompositions can be applied using almost any base binary classifier or regressor. In our experiments, we have selected in those cases SVMs, given that they are suggested by many of the authors of the different works analysed. Starting with naïve approaches, the following methods were considered: 1) C-Support Vector Classifier (C-SVC) with *OneVsOne* and *OneVsAll* decompositions (SVC1V1 and SVC1VA), because they are the two main approaches when applying SVM to multiclass problems [59]. Although these methods consider binary decompositions, they have been included in the nominal classification group, given that they do not take the class order into account. 2) Support Vector Regression (SVR) applied to a modified dataset where the target variable $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$ is mapped to the real values $\{0, 1/(Q-1), 2/(Q-1), \ldots, 1\}$. The concrete regression model considered is the ϵ-SVR [52]. 3) Cost-Sensitive SVC (CSSVC), which is a C-SVC [59] with the *OneVsAll* decomposition, where costs are included as different weights [127] for the negative class of each decomposition. Absolute costs are considered for the errors.

Regarding the ordinal binary decompositions, the methods considered are the following: 1) The *OrderedPartitions* decomposition was applied to the C-SVC classification algorithm (SVMOP), but including different weights as proposed by Waegeman et al. [65]. However, given the problem of possible ambiguities recognised by the authors, probability estimates are obtained following the method presented in [128], considering the fusion of probabilities presented by Frank and Hall [63] (equation (1)). 2) The neural network model proposed in [71] (NNOP). This model considers the *OrderedPartitions* coding scheme for the labels and a rule for decisions based on the first node whose output is higher than a predefined threshold ($T = 0.5$, in our experiments). We consider then the mean square error function over the outputs and the iRProp+ algorithm [129] to optimise the parameters. 3) Finally, the single model ordinal ELM presented in [73] (ELMOP).

The threshold models considered are the following: 1) The POM [83], with the *logit* link function (the most popular one). 2) A neural network approach based on the POM (NNPOM), such as the one proposed by Mathieson [18]. By considering corresponding probabilities, the cross entropy function is optimised by the iRProp+ algorithm [129]. Threshold constraints are satisfied by substituting the set of parameters $\{b_1, b_2, \ldots, b_Q\}$ by $\{\alpha_1, \alpha_1+\alpha_2^2, \ldots, \alpha_1+\alpha_2^2+\ldots+\alpha_Q^2\}$, which allows unconstrained optimisation of $\{\alpha_1, \ldots, \alpha_Q\}$. 3) Ordinal support vector formulations of Chu and Keerthi [29], including both explicitly and implicitly constrained alternatives (SVOREX and SVORIM). 4) KDLOR algorithm presented in [31]. 5) The GPOR method [30] including automatic relevance determination, as proposed by the authors. 6) The reduction from ordinal regression to binary classification applied to SVMs (REDSVM) was also considered. The configuration used was the identity coding matrix, the absolute cost matrix and the standard binary soft-margin SVM, as proposed in [41]. 7) Finally, the ORBoost method with all margins and left-right margins [107] (ORBALL and ORBLR). As proposed by the authors, the total number of ensemble members is set to $T = 2000$, and normalised sigmoid functions are used as the base classifier, where the smoothness parameter is $\gamma = 4$ [107].

TABLE 5
Different algorithms considered for the experiments

Abbr.	Short description
	Naïve approaches
SVC1V1	Support Vector Classifier with *OneVsOne* [59]
SVC1VA	Support Vector Classifier with *OneVsAll* [59]
SVR	Support Vector Machines for regression [52]
CSSVC	Cost-Sensitive Support Vector Classifier (CSSVC) [59]
	Ordinal Binary decompositions
SVMOP	Support Vector Machines with *OrderedPartitions* [63], [65]
NNOP	Neural Network with *OrderedPartitions* [71]
ELMOP	Extreme Learning Machine with *OrderedPartitions* [73]
	Threshold models
POM	Proportional Odds Model [80]
NNPOM	Neural Network based on Proportional Odd Model [18]
SVOREX	Support Vector Ordinal Regression with Explicit Constraints [29]
SVORIM	Support Vector Ordinal Regression with Implicit Constraints [29]
KDLOR	Kernel Discriminant Learning for Ordinal Regression [31]
GPOR	Gaussian Processes for Ordinal Regression [30]
REDSVM	Reduction applied to Support Vector Machines [41]
ORBALL	Ordinal Regression Boosting with All margins [107]
ORBLR	Ordinal Regression Boosting with Left-Right margins [107]

4.1.3 Performance evaluation and model selection

Different measures can be considered for evaluating ordinal regression models [119], [130], [131]. However, the most common ones are the Mean Zero-one Error (MZE) and the Mean Absolute Error (MAE). MZE is the error rate of the classifier:

$$MZE = \frac{1}{N}\sum_{i=1}^{N}[\![y_i^* \neq y_i]\!] = 1 - Acc,$$

where y_i is the true label, y_i^* is the predicted label and Acc is the accuracy of classifier. MZE values range from 0 to 1. It is related to global performance, but without considering different errors with regards to ordering.

The MAE is the average deviation in absolute value of the predicted rank ($\mathcal{O}(y_i^*)$) from the true one ($\mathcal{O}(y_i)$) [131]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|.$$

MAE values range from 0 to $Q-1$ (maximum deviation in number of categories). In this way, MZE considers a zero-one loss for misclassification, while MAE uses an absolute cost. We consider these costs for evaluating the datasets because they are most common (for example, see [29]–[31], just to cite some of them).

TABLE 6
Test MZE results for each dataset and method, including the average over all the splits and the standard deviation.

Discretised regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	KDLOR	GPOR	REDSVM	ORBALL	ORBLR
P5	.546$_{.091}$	.542$_{.075}$	*.492$_{.090}$*	.533$_{.063}$	.521$_{.083}$	.646$_{.109}$	.652$_{.088}$	**.485$_{.118}$**	.646$_{.085}$	.508$_{.095}$	.498$_{.092}$	.512$_{.081}$	.515$_{.096}$	.506$_{.080}$	.562$_{.067}$	.560$_{.072}$
M5	.428$_{.060}$	.452$_{.061}$	.431$_{.056}$	.458$_{.059}$	.414$_{.063}$	.400$_{.066}$	.402$_{.074}$	.394$_{.065}$	.431$_{.073}$	.432$_{.058}$	.416$_{.067}$	.425$_{.064}$	.403$_{.054}$	.400$_{.064}$	*.389$_{.069}$*	**.377$_{.071}$**
H5	.353$_{.034}$	.374$_{.027}$	.351$_{.031}$	.372$_{.025}$	.351$_{.030}$	.358$_{.026}$	.361$_{.027}$	.355$_{.018}$	.386$_{.027}$	.329$_{.022}$	.325$_{.027}$	.362$_{.027}$	**.310$_{.029}$**	*.324$_{.029}$*	.328$_{.025}$	.330$_{.030}$
S5	.107$_{.013}$	.112$_{.017}$	.120$_{.016}$	.107$_{.016}$	.116$_{.020}$	.122$_{.018}$	.112$_{.014}$	.370$_{.016}$	.119$_{.015}$	.113$_{.014}$	.113$_{.014}$	.113$_{.013}$	.109$_{.015}$	.111$_{.018}$	**.101$_{.014}$**	*.103$_{.012}$*
A5	.512$_{.010}$	.527$_{.008}$	.530$_{.007}$	.524$_{.009}$	**.507$_{.010}$**	.520$_{.009}$	.521$_{.007}$	.539$_{.005}$	.518$_{.009}$	.512$_{.006}$	.524$_{.008}$	.546$_{.009}$	*.509$_{.007}$*	.524$_{.008}$	.529$_{.008}$	.514$_{.009}$
B5	.453$_{.064}$	.539$_{.044}$	.357$_{.056}$	.546$_{.032}$	.431$_{.035}$	.705$_{.071}$	.545$_{.046}$	*.271$_{.024}$*	.765$_{.023}$	.273$_{.025}$	.276$_{.029}$	.315$_{.048}$	**.267$_{.017}$**	.283$_{.038}$	.329$_{.028}$	.329$_{.028}$
BB5	.709$_{.021}$	.690$_{.020}$	.664$_{.020}$	.684$_{.016}$	.693$_{.019}$	.780$_{.014}$	.735$_{.026}$	*.639$_{.026}$*	.786$_{.013}$	.652$_{.024}$	.646$_{.024}$	.657$_{.024}$	**.585$_{.029}$**	.644$_{.024}$	.673$_{.018}$	.673$_{.018}$
C5	.421$_{.016}$	.470$_{.017}$	.424$_{.035}$	.463$_{.020}$	.413$_{.018}$	.426$_{.017}$	.461$_{.030}$	*.381$_{.014}$*	.453$_{.036}$	.390$_{.028}$	.397$_{.031}$	.429$_{.027}$	**.377$_{.014}$**	.397$_{.038}$	.458$_{.026}$	.457$_{.025}$
CC5	.380$_{.032}$	.425$_{.023}$	.375$_{.018}$	.421$_{.021}$	.370$_{.024}$	.406$_{.017}$	.457$_{.046}$	.342$_{.019}$	.415$_{.021}$	.343$_{.016}$	*.338$_{.014}$*	.387$_{.028}$	**.325$_{.017}$**	.341$_{.019}$	.434$_{.031}$	.432$_{.029}$
CH5	.519$_{.021}$	.542$_{.022}$	.527$_{.020}$	.546$_{.023}$	.516$_{.023}$	.520$_{.016}$	.547$_{.019}$	**.492$_{.010}$**	.527$_{.032}$	.496$_{.011}$	.499$_{.011}$	.510$_{.017}$	.497$_{.017}$	*.494$_{.016}$*	.515$_{.012}$	.513$_{.011}$
CE5	.564$_{.018}$	.576$_{.016}$	.587$_{.015}$	.587$_{.021}$	.567$_{.017}$	.587$_{.023}$	.595$_{.019}$	.558$_{.010}$	.596$_{.023}$	*.555$_{.013}$*	.556$_{.012}$	.568$_{.018}$	**.535$_{.009}$**	.559$_{.013}$	.592$_{.016}$	.591$_{.016}$
CEE5	.564$_{.016}$	.585$_{.017}$	.573$_{.016}$	.585$_{.018}$	.560$_{.020}$	.600$_{.016}$	.599$_{.016}$	.584$_{.017}$	.634$_{.020}$	.553$_{.018}$	.555$_{.016}$	*.552$_{.016}$*	**.539$_{.009}$**	.553$_{.013}$	.583$_{.011}$	.582$_{.010}$
A$_{D5}$	.463$_{.148}$	.486$_{.144}$	.453$_{.144}$	.486$_{.145}$	.455$_{.144}$	.506$_{.178}$	.499$_{.161}$	.451$_{.114}$	.523$_{.183}$	.430$_{.148}$	.428$_{.148}$	.448$_{.144}$	**.414$_{.142}$**	*.428$_{.148}$*	.458$_{.155}$	.455$_{.154}$
P10	.775$_{.069}$	.758$_{.064}$	.769$_{.091}$	.758$_{.063}$	.763$_{.079}$	.802$_{.075}$	.794$_{.081}$	*.719$_{.076}$*	.850$_{.073}$	.783$_{.076}$	.735$_{.095}$	.756$_{.086}$	.779$_{.087}$	**.717$_{.102}$**	.760$_{.091}$	.760$_{.095}$
M10	**.597$_{.055}$**	.641$_{.062}$	.634$_{.065}$	.634$_{.047}$	.643$_{.047}$	.621$_{.058}$	.610$_{.054}$	.642$_{.059}$	.653$_{.054}$	.624$_{.062}$	.629$_{.054}$	.651$_{.065}$	.642$_{.056}$	.618$_{.051}$	.606$_{.055}$	*.603$_{.057}$*
H10	.597$_{.025}$	.636$_{.028}$	.590$_{.035}$	.623$_{.019}$	.589$_{.031}$	.584$_{.033}$	.599$_{.029}$	.592$_{.032}$	.606$_{.033}$	.574$_{.033}$	.557$_{.029}$	.599$_{.038}$	**.542$_{.026}$**	.548$_{.032}$	.551$_{.029}$	*.545$_{.031}$*
S10	**.206$_{.022}$**	.218$_{.019}$	.231$_{.019}$	.215$_{.020}$	*.212$_{.021}$*	.249$_{.033}$	.238$_{.018}$	.561$_{.019}$	.257$_{.025}$	.235$_{.023}$	.234$_{.021}$	.230$_{.019}$	.249$_{.019}$	.230$_{.021}$	.232$_{.025}$	.234$_{.024}$
A10	.713$_{.008}$	.733$_{.012}$	.739$_{.005}$	.725$_{.008}$	.718$_{.008}$	.725$_{.010}$	.731$_{.008}$	.743$_{.005}$	.719$_{.008}$	*.709$_{.008}$*	.733$_{.008}$	.744$_{.009}$	**.708$_{.005}$**	.728$_{.010}$	.732$_{.007}$	.713$_{.008}$
B10	.755$_{.033}$	.766$_{.020}$	.588$_{.052}$	.763$_{.020}$	.704$_{.047}$	.851$_{.044}$	.754$_{.044}$	**.472$_{.019}$**	.881$_{.013}$	.506$_{.020}$	.494$_{.021}$	.524$_{.047}$	*.485$_{.016}$*	.487$_{.022}$	.560$_{.023}$	.559$_{.023}$
BB10	.858$_{.010}$	.860$_{.011}$	.825$_{.024}$	.848$_{.009}$	.861$_{.015}$	.895$_{.006}$	.869$_{.014}$	*.805$_{.014}$*	.894$_{.006}$	.814$_{.017}$	.811$_{.018}$	.816$_{.019}$	**.777$_{.014}$**	.815$_{.017}$	.831$_{.012}$	.832$_{.013}$
C10	.643$_{.016}$	.677$_{.019}$	.634$_{.032}$	.677$_{.016}$	.631$_{.016}$	.634$_{.011}$	.672$_{.024}$	**.580$_{.014}$**	.657$_{.035}$	.598$_{.027}$	.607$_{.024}$	.644$_{.028}$	*.588$_{.017}$*	.601$_{.027}$	.654$_{.015}$	.649$_{.014}$
CC10	.603$_{.013}$	.647$_{.020}$	.585$_{.032}$	.651$_{.027}$	.591$_{.019}$	.605$_{.021}$	.645$_{.024}$	*.534$_{.012}$*	.628$_{.021}$	.549$_{.030}$	.546$_{.021}$	.611$_{.034}$	**.527$_{.013}$**	.548$_{.020}$	.642$_{.024}$	.631$_{.019}$
CH10	.732$_{.019}$	.744$_{.016}$	.725$_{.013}$	.744$_{.013}$	.733$_{.022}$	.723$_{.015}$	.746$_{.017}$	.704$_{.011}$	.719$_{.022}$	*.702$_{.012}$*	.707$_{.015}$	.717$_{.018}$	**.696$_{.014}$**	.706$_{.018}$	.713$_{.011}$	.707$_{.010}$
CE10	.770$_{.012}$	.780$_{.015}$	.770$_{.009}$	.779$_{.011}$	.768$_{.019}$	.778$_{.013}$	.778$_{.012}$	*.753$_{.012}$*	.786$_{.014}$	.755$_{.014}$	.756$_{.014}$	.776$_{.014}$	**.736$_{.007}$**	.758$_{.018}$	.778$_{.010}$	.772$_{.009}$
CEE10	.778$_{.022}$	.782$_{.014}$	.767$_{.014}$	.782$_{.013}$	.771$_{.013}$	.785$_{.009}$	.788$_{.009}$	.780$_{.008}$	.813$_{.012}$	*.758$_{.009}$*	.759$_{.012}$	.761$_{.015}$	**.749$_{.010}$**	.762$_{.018}$	.774$_{.007}$	.771$_{.007}$
A$_{D10}$	.669$_{.168}$	.687$_{.163}$	.655$_{.158}$	.683$_{.162}$	.665$_{.164}$	.688$_{.171}$	.685$_{.162}$	.657$_{.108}$	.705$_{.173}$	.634$_{.160}$	.631$_{.160}$	.653$_{.159}$	**.623$_{.155}$**	*.626$_{.161}$*	.653$_{.160}$	.648$_{.159}$
A$_{D}$	.566$_{.188}$	.586$_{.182}$	.554$_{.180}$	.585$_{.181}$	.560$_{.185}$	.597$_{.194}$	.592$_{.185}$	.554$_{.151}$	.614$_{.197}$	.532$_{.184}$	.530$_{.183}$	.550$_{.181}$	**.519$_{.180}$**	*.527$_{.182}$*	.555$_{.184}$	.551$_{.182}$
Ordinal regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	KDLOR	GPOR	REDSVM	ORBALL	ORBLR
CL	*.278$_{.126}$*	.278$_{.110}$	.317$_{.080}$	**.261$_{.136}$**	.367$_{.102}$	.283$_{.125}$	.422$_{.218}$	.383$_{.170}$	.356$_{.143}$	.356$_{.129}$	.383$_{.117}$	.339$_{.155}$	.394$_{.093}$	.372$_{.121}$	.356$_{.129}$	.367$_{.119}$
PA	.307$_{.119}$	.330$_{.107}$	.337$_{.129}$	.322$_{.114}$	.319$_{.091}$	**.237$_{.116}$**	.400$_{.184}$	.504$_{.154}$	.344$_{.178}$	.348$_{.119}$	.344$_{.122}$	.326$_{.116}$	.478$_{.178}$	.326$_{.116}$	.300$_{.121}$	*.270$_{.116}$*
SS	*.359$_{.095}$*	.403$_{.145}$	.387$_{.113}$	.395$_{.134}$	.403$_{.094}$	.395$_{.116}$	.444$_{.162}$	.618$_{.152}$	.500$_{.131}$	.374$_{.127}$	.372$_{.129}$	.392$_{.125}$	.549$_{.100}$	.379$_{.132}$	.364$_{.124}$	**.351$_{.121}$**
SU	**.221$_{.121}$**	.279$_{.158}$	*.244$_{.103}$*	.269$_{.156}$	.267$_{.116}$	.295$_{.113}$	.387$_{.129}$	.651$_{.142}$	.390$_{.143}$	.264$_{.114}$	.269$_{.114}$	.264$_{.108}$	.356$_{.162}$	.269$_{.119}$	.297$_{.098}$	.303$_{.112}$
TA	.439$_{.059}$	.448$_{.065}$	**.396$_{.073}$**	.428$_{.072}$	.456$_{.057}$	.415$_{.060}$	.436$_{.079}$	.496$_{.077}$	.453$_{.090}$	.411$_{.069}$	.401$_{.072}$	.433$_{.055}$	.672$_{.041}$	*.399$_{.068}$*	.403$_{.057}$	.414$_{.064}$
NT	.035$_{.027}$	.041$_{.029}$	.044$_{.024}$	.038$_{.025}$	.041$_{.027}$	.035$_{.022}$	.057$_{.024}$	*.028$_{.022}$*	.033$_{.025}$	.034$_{.023}$	.034$_{.023}$	**.026$_{.020}$**	.034$_{.024}$	.032$_{.023}$	.042$_{.029}$	.041$_{.030}$
BS	.028$_{.015}$	.033$_{.012}$	.165$_{.027}$	.031$_{.013}$	.034$_{.014}$	.039$_{.013}$	.087$_{.023}$	.094$_{.019}$	.062$_{.048}$	*.002$_{.006}$*	.002$_{.006}$	.160$_{.031}$	.034$_{.012}$	**.001$_{.004}$**	.032$_{.016}$	.032$_{.018}$
SW	*.422$_{.034}$*	.437$_{.035}$	.435$_{.030}$	.429$_{.032}$	.424$_{.035}$	.423$_{.035}$	.426$_{.025}$	.432$_{.030}$	.456$_{.032}$	.432$_{.030}$	.431$_{.027}$	.514$_{.028}$	**.422$_{.031}$**	.429$_{.027}$	.439$_{.032}$	.437$_{.032}$
CA	*.006$_{.005}$*	.014$_{.006}$	.027$_{.006}$	.014$_{.006}$	**.003$_{.004}$**	.026$_{.013}$	.159$_{.015}$	.843$_{.306}$	.106$_{.022}$	.012$_{.006}$	.012$_{.005}$	.047$_{.010}$	.037$_{.009}$	.012$_{.004}$	.012$_{.006}$	.011$_{.005}$
BO	*.429$_{.042}$*	.431$_{.060}$	.458$_{.072}$	.433$_{.060}$	.447$_{.098}$	.533$_{.093}$	.567$_{.135}$	.656$_{.161}$	.618$_{.143}$	.453$_{.062}$	.451$_{.074}$	.473$_{.093}$	**.422$_{.032}$**	.436$_{.055}$	.458$_{.091}$	.453$_{.095}$
TO	.050$_{.026}$	.055$_{.028}$	.068$_{.037}$	.054$_{.028}$	.072$_{.028}$	.062$_{.031}$	.071$_{.029}$	.711$_{.026}$	.062$_{.032}$	**.020$_{.014}$**	*.020$_{.014}$*	.107$_{.032}$	.046$_{.022}$	.023$_{.012}$	.052$_{.025}$	.053$_{.026}$
EU	.360$_{.029}$	.445$_{.026}$	.361$_{.027}$	.435$_{.026}$	*.350$_{.026}$*	.423$_{.036}$	.428$_{.029}$	.851$_{.016}$	.462$_{.042}$	.364$_{.027}$	.361$_{.033}$	.372$_{.028}$	**.314$_{.034}$**	.362$_{.035}$	.380$_{.029}$	.375$_{.033}$
LE	.368$_{.025}$	.372$_{.020}$	.375$_{.025}$	**.367$_{.019}$**	*.367$_{.025}$*	.375$_{.026}$	.367$_{.022}$	.377$_{.028}$	.382$_{.024}$	.375$_{.022}$	.380$_{.018}$	.458$_{.031}$	.388$_{.030}$	.373$_{.024}$	.391$_{.029}$	.387$_{.024}$
AU	**.246$_{.058}$**	.260$_{.058}$	.321$_{.067}$	.265$_{.061}$	*.258$_{.042}$*	.389$_{.060}$	.381$_{.061}$	.533$_{.194}$	.550$_{.077}$	.316$_{.055}$	.323$_{.069}$	.301$_{.067}$	.389$_{.073}$	.317$_{.070}$	.294$_{.055}$	.299$_{.053}$
WI	.352$_{.023}$	.361$_{.024}$	.372$_{.019}$	.362$_{.023}$	.358$_{.020}$	.402$_{.018}$	.397$_{.016}$	.403$_{.015}$	.401$_{.022}$	.373$_{.019}$	.373$_{.020}$	.350$_{.019}$	.394$_{.015}$	.373$_{.020}$	*.334$_{.021}$*	**.333$_{.019}$**
ES	.307$_{.037}$	.328$_{.027}$	.296$_{.034}$	.320$_{.026}$	.289$_{.032}$	.308$_{.039}$	.302$_{.039}$	.295$_{.034}$	.341$_{.129}$	.290$_{.026}$	**.284$_{.032}$**	.355$_{.034}$	*.287$_{.031}$*	.287$_{.030}$	.323$_{.022}$	.325$_{.029}$
ER	.736$_{.025}$	.825$_{.026}$	.750$_{.023}$	.801$_{.029}$	.745$_{.023}$	**.708$_{.017}$**	.746$_{.019}$	.744$_{.021}$	.727$_{.028}$	.714$_{.026}$	.751$_{.021}$	.805$_{.031}$	*.712$_{.027}$*	.751$_{.019}$	.760$_{.021}$	.727$_{.025}$
A$_{OR}$	**.291$_{.187}$**	.314$_{.202}$	.315$_{.175}$	.307$_{.198}$	.306$_{.188}$	.315$_{.188}$	.357$_{.179}$	.507$_{.234}$	.367$_{.199}$	*.302$_{.189}$*	.305$_{.194}$	.337$_{.189}$	.349$_{.210}$	.302$_{.193}$	.308$_{.189}$	.305$_{.184}$

The best result for each dataset is in bold face and the second one in italics

Multiple random splits of the datasets were considered. For discretised regression datasets, 20 random splits were done and the number of training and test patterns were those suggested in [30]. For real ordinal regression problems, 30 random stratified splits with 75% and 25% of the patterns in the training and test sets were considered, respectively (as suggested in [132]). All the partitions were the same for all the methods, and one model was trained and evaluated for each split. Then, MZE or MAE were obtained, and the computational times needed for training each model were also gathered.

All SVM classifiers or regressors were run using the implementations available in the `libsvm` library (version 3.0) [127]. The `mnrfit` function of Matlab was used for training the POM model. The authors of GPOR, SVOREX, SVORIM, RED-SVM and ORBoost provide publicly available software implementations of their methods[3]. All the experiments were run using a common Matlab framework, with an Intel(R) Xeon(R) CPU E5405 at 2.00GHz with 8GB of RAM. This framework is available, together with all the datasets and partitions, the individual results and the detailed results of the statistical tests, on the website associated with this paper[4].

It is very important to consider a proper model selection process to assure a fair comparison. In this sense, all model hyperparameters were selected by using a nested five fold cross-validation over the training set. Once the lowest cross-validation error alternative was obtained,

3. GPOR (http://www.gatsby.ucl.ac.uk/~huwei/ordinalregression.html), SVOREX and SVORIM (http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm), ORBoost (http://www.work.caltech.edu/~htlin/program/orensemble/) and RED-SVM (http://home.caltech.edu/~htlin/program/libsvm/)

4. http://www.uco.es/grupos/ayrna/orreview

TABLE 7
Test MAE results for each dataset and method, including the average over all the splits and the standard deviation.

Discretised regression datasets ($Mean_{SD}$)																
Dataset	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	KDLOR	GPOR	REDSVM	ORBALL	ORBLR
P5	$0.79_{0.13}$	$0.75_{0.18}$	$0.66_{0.14}$	$0.74_{0.15}$	$0.69_{0.16}$	$0.91_{0.21}$	$1.12_{0.26}$	$0.70_{0.20}$	$0.96_{0.21}$	$0.66_{0.12}$	$\mathbf{0.63_{0.11}}$	$0.67_{0.19}$	$0.65_{0.13}$	*0.64*$_{0.14}$	$0.73_{0.10}$	$0.73_{0.10}$
M5	$0.47_{0.08}$	$0.55_{0.08}$	$0.48_{0.09}$	$0.55_{0.10}$	$0.46_{0.08}$	$0.46_{0.07}$	$0.45_{0.09}$	$0.43_{0.08}$	$0.49_{0.07}$	$0.47_{0.07}$	$0.44_{0.10}$	$0.49_{0.10}$	$0.44_{0.07}$	$0.46_{0.09}$	*0.41*$_{0.08}$	$\mathbf{0.40_{0.08}}$
H5	$0.41_{0.05}$	$0.44_{0.04}$	$0.38_{0.04}$	$0.44_{0.04}$	$0.40_{0.04}$	$0.41_{0.03}$	$0.42_{0.04}$	$0.40_{0.02}$	$0.45_{0.05}$	$0.36_{0.03}$	$0.36_{0.03}$	$0.39_{0.05}$	$\mathbf{0.34_{0.04}}$	*0.36*$_{0.03}$	$0.36_{0.03}$	$0.36_{0.04}$
S5	$0.11_{0.01}$	$0.11_{0.01}$	$0.12_{0.02}$	$0.11_{0.02}$	$0.11_{0.02}$	$0.13_{0.02}$	$0.11_{0.01}$	$0.39_{0.02}$	$0.12_{0.02}$	$0.11_{0.01}$	$0.11_{0.02}$	$0.11_{0.01}$	$0.11_{0.01}$	$0.11_{0.02}$	$\mathbf{0.10_{0.01}}$	*0.10*$_{0.01}$
A5	$0.71_{0.02}$	$0.79_{0.02}$	$0.66_{0.01}$	$0.78_{0.02}$	$0.67_{0.01}$	$0.66_{0.01}$	$0.67_{0.01}$	$0.69_{0.01}$	$0.70_{0.01}$	$0.66_{0.01}$	$\mathbf{0.65_{0.01}}$	$0.76_{0.02}$	$0.68_{0.01}$	*0.65*$_{0.01}$	$0.67_{0.01}$	$0.69_{0.01}$
B5	$0.50_{0.08}$	$0.67_{0.09}$	$0.36_{0.05}$	$0.67_{0.07}$	$0.47_{0.04}$	$1.13_{0.19}$	$0.69_{0.10}$	*0.28*$_{0.03}$	$1.32_{0.19}$	$0.28_{0.03}$	$0.28_{0.03}$	$0.33_{0.06}$	$\mathbf{0.27_{0.02}}$	$0.28_{0.03}$	$0.34_{0.03}$	$0.33_{0.03}$
BB5	$1.18_{0.07}$	$1.18_{0.05}$	$0.94_{0.08}$	$1.19_{0.07}$	$1.09_{0.07}$	$1.40_{0.06}$	$1.30_{0.15}$	$0.98_{0.09}$	$1.47_{0.09}$	$0.94_{0.08}$	$0.92_{0.07}$	$0.95_{0.08}$	$\mathbf{0.83_{0.08}}$	$0.93_{0.07}$	$0.89_{0.05}$	*0.89*$_{0.05}$
C5	$0.51_{0.03}$	$0.59_{0.05}$	$0.49_{0.03}$	$0.57_{0.04}$	$0.50_{0.03}$	$0.50_{0.03}$	$0.55_{0.04}$	*0.43*$_{0.02}$	$0.56_{0.06}$	$0.45_{0.05}$	$0.45_{0.04}$	$0.52_{0.06}$	$\mathbf{0.43_{0.02}}$	$0.44_{0.04}$	$0.52_{0.04}$	$0.52_{0.04}$
CC5	$0.43_{0.05}$	$0.50_{0.03}$	$0.40_{0.02}$	$0.50_{0.03}$	$0.42_{0.04}$	$0.49_{0.20}$	$0.53_{0.06}$	$0.38_{0.03}$	$0.48_{0.03}$	$0.37_{0.02}$	$0.37_{0.01}$	$0.44_{0.04}$	$\mathbf{0.35_{0.02}}$	*0.37*$_{0.02}$	$0.48_{0.05}$	$0.48_{0.05}$
CH5	$0.66_{0.04}$	$0.75_{0.04}$	$0.63_{0.03}$	$0.75_{0.04}$	$0.64_{0.03}$	$0.65_{0.03}$	$0.70_{0.03}$	*0.59*$_{0.01}$	$0.67_{0.05}$	$0.60_{0.02}$	$0.60_{0.02}$	$0.65_{0.04}$	$0.62_{0.03}$	$\mathbf{0.59_{0.02}}$	$0.64_{0.02}$	$0.64_{0.02}$
CE5	$0.79_{0.03}$	$0.86_{0.04}$	$0.76_{0.03}$	$0.87_{0.05}$	$0.77_{0.03}$	$0.81_{0.03}$	$0.80_{0.03}$	$0.73_{0.02}$	$0.87_{0.05}$	$0.74_{0.03}$	$\mathbf{0.73_{0.02}}$	$0.79_{0.04}$	$0.74_{0.02}$	*0.73*$_{0.02}$	$0.79_{0.03}$	$0.79_{0.03}$
CEE5	$0.81_{0.03}$	$0.88_{0.08}$	$0.74_{0.02}$	$0.86_{0.04}$	$0.79_{0.04}$	$0.87_{0.04}$	$0.83_{0.03}$	$0.78_{0.03}$	$0.93_{0.04}$	$0.74_{0.03}$	*0.72*$_{0.03}$	$0.74_{0.03}$	$0.74_{0.02}$	$\mathbf{0.72_{0.03}}$	$0.73_{0.01}$	$0.73_{0.01}$
A_{D5}	$0.61_{0.27}$	$0.67_{0.27}$	$0.55_{0.22}$	$0.67_{0.27}$	$0.58_{0.25}$	$0.70_{0.35}$	$0.68_{0.32}$	$0.57_{0.21}$	$0.75_{0.38}$	$0.53_{0.23}$	*0.52*$_{0.23}$	$0.57_{0.23}$	$\mathbf{0.52_{0.22}}$	$0.52_{0.23}$	$0.56_{0.23}$	$0.56_{0.23}$
P10	$1.88_{0.40}$	$1.88_{0.35}$	$1.43_{0.22}$	$1.82_{0.43}$	$1.46_{0.30}$	$1.77_{0.37}$	$2.08_{0.53}$	$1.54_{0.29}$	$2.20_{0.80}$	$1.44_{0.23}$	*1.39*$_{0.19}$	$1.43_{0.26}$	$1.55_{0.24}$	$\mathbf{1.32_{0.19}}$	$1.53_{0.22}$	$1.51_{0.26}$
M10	$1.05_{0.14}$	$1.14_{0.15}$	$1.03_{0.15}$	$1.11_{0.16}$	$0.99_{0.09}$	$0.94_{0.16}$	*0.92*$_{0.12}$	$0.92_{0.11}$	$1.12_{0.15}$	$0.96_{0.12}$	$0.92_{0.13}$	$1.09_{0.20}$	$0.97_{0.11}$	$0.92_{0.11}$	$\mathbf{0.91_{0.12}}$	$0.92_{0.10}$
H10	$0.89_{0.08}$	$1.07_{0.09}$	$0.80_{0.06}$	$1.03_{0.09}$	$0.88_{0.06}$	$0.87_{0.07}$	$0.90_{0.06}$	$0.88_{0.05}$	$0.95_{0.11}$	$0.78_{0.06}$	$0.77_{0.06}$	$0.84_{0.06}$	*0.76*$_{0.05}$	$0.76_{0.07}$	$0.76_{0.06}$	$\mathbf{0.75_{0.07}}$
S10	$\mathbf{0.22_{0.02}}$	$0.22_{0.02}$	$0.24_{0.02}$	$0.23_{0.02}$	*0.22*$_{0.02}$	$0.27_{0.04}$	$0.25_{0.02}$	$0.82_{0.03}$	$0.27_{0.03}$	$0.23_{0.02}$	$0.24_{0.02}$	$0.24_{0.02}$	$0.25_{0.02}$	$0.23_{0.02}$	$0.24_{0.02}$	$0.24_{0.02}$
A10	$1.56_{0.03}$	$1.73_{0.06}$	$1.38_{0.02}$	$1.74_{0.03}$	$1.43_{0.03}$	$1.38_{0.02}$	$1.39_{0.02}$	$1.45_{0.01}$	$1.49_{0.02}$	$1.47_{0.02}$	*1.36*$_{0.02}$	$1.64_{0.04}$	$1.49_{0.02}$	$\mathbf{1.36_{0.02}}$	$1.40_{0.02}$	$1.49_{0.03}$
B10	$1.47_{0.28}$	$1.62_{0.09}$	$0.72_{0.11}$	$1.64_{0.16}$	$0.98_{0.06}$	$2.24_{0.58}$	$1.46_{0.29}$	$\mathbf{0.54_{0.03}}$	$2.87_{0.43}$	$0.59_{0.05}$	$0.57_{0.03}$	$0.63_{0.10}$	$0.57_{0.03}$	*0.56*$_{0.05}$	$0.67_{0.04}$	$0.67_{0.04}$
BB10	$2.51_{0.10}$	$2.60_{0.18}$	$1.91_{0.12}$	$2.58_{0.14}$	$2.27_{0.19}$	$2.91_{0.10}$	$2.56_{0.30}$	$1.98_{0.14}$	$3.00_{0.13}$	$1.92_{0.14}$	$1.88_{0.13}$	$1.92_{0.12}$	*1.83*$_{0.18}$	$1.88_{0.13}$	$\mathbf{1.83_{0.10}}$	$1.84_{0.10}$
C10	$1.15_{0.06}$	$1.30_{0.11}$	$0.99_{0.05}$	$1.28_{0.13}$	$1.06_{0.05}$	$1.06_{0.06}$	$1.18_{0.06}$	$\mathbf{0.87_{0.03}}$	$1.16_{0.10}$	$0.93_{0.05}$	*0.90*$_{0.03}$	$1.08_{0.10}$	$0.92_{0.04}$	$0.90_{0.04}$	$1.04_{0.06}$	$1.03_{0.05}$
CC10	$0.94_{0.05}$	$1.14_{0.06}$	$0.83_{0.05}$	$1.12_{0.07}$	$0.89_{0.05}$	$0.92_{0.05}$	$1.04_{0.10}$	*0.75*$_{0.04}$	$1.05_{0.12}$	$0.79_{0.06}$	$0.76_{0.05}$	$0.97_{0.10}$	$\mathbf{0.74_{0.03}}$	$0.76_{0.03}$	$0.96_{0.09}$	$0.93_{0.06}$
CH10	$1.46_{0.06}$	$1.67_{0.08}$	$1.27_{0.04}$	$1.63_{0.07}$	$1.37_{0.06}$	$1.34_{0.06}$	$1.43_{0.05}$	$\mathbf{1.23_{0.04}}$	$1.40_{0.11}$	$1.31_{0.05}$	$1.25_{0.04}$	$1.41_{0.10}$	$1.30_{0.06}$	*1.25*$_{0.04}$	$1.31_{0.03}$	$1.31_{0.04}$
CE10	$1.77_{0.09}$	$1.91_{0.12}$	$1.55_{0.04}$	$1.92_{0.17}$	$1.64_{0.06}$	$1.72_{0.07}$	$1.65_{0.06}$	$1.54_{0.06}$	$1.86_{0.11}$	$1.65_{0.07}$	*1.51*$_{0.04}$	$1.75_{0.12}$	$1.61_{0.06}$	$\mathbf{1.50_{0.05}}$	$1.64_{0.07}$	$1.65_{0.07}$
CEE10	$1.76_{0.05}$	$1.92_{0.06}$	$1.53_{0.05}$	$1.89_{0.06}$	$1.64_{0.06}$	$1.77_{0.07}$	$1.72_{0.06}$	$1.64_{0.04}$	$2.00_{0.14}$	$1.62_{0.08}$	$\mathbf{1.50_{0.04}}$	$1.61_{0.09}$	$1.64_{0.08}$	*1.51*$_{0.06}$	$1.51_{0.03}$	$1.51_{0.03}$
A_{D10}	$1.39_{0.59}$	$1.52_{0.60}$	$1.14_{0.46}$	$1.50_{0.59}$	$1.24_{0.52}$	$1.43_{0.71}$	$1.38_{0.60}$	$1.18_{0.44}$	$1.61_{0.81}$	$1.14_{0.50}$	*1.09*$_{0.47}$	$1.22_{0.50}$	$1.14_{0.50}$	$\mathbf{1.08_{0.47}}$	$1.15_{0.47}$	$1.15_{0.47}$
A_D	$1.00_{0.60}$	$1.09_{0.62}$	$0.85_{0.46}$	$1.08_{0.62}$	$0.91_{0.52}$	$1.07_{0.66}$	$1.03_{0.59}$	$0.87_{0.46}$	$1.18_{0.76}$	$0.84_{0.49}$	*0.80*$_{0.46}$	$0.89_{0.51}$	$0.83_{0.49}$	$\mathbf{0.80_{0.46}}$	$0.85_{0.47}$	$0.85_{0.47}$
Ordinal regression datasets ($Mean_{SD}$)																
Datasets	SCV1V1	SVC1VA	SVR	CSSVC	SVMOP	NNOP	ELMOP	POM	NNPOM	SVOREX	SVORIM	KDLOR	GPOR	REDSVM	ORBALL	ORBLR
CL	$0.52_{0.22}$	$0.46_{0.19}$	$\mathbf{0.38_{0.15}}$	$0.46_{0.23}$	$0.50_{0.18}$	$0.46_{0.25}$	$0.52_{0.28}$	$0.53_{0.25}$	$0.48_{0.23}$	$0.48_{0.13}$	$0.52_{0.17}$	$0.52_{0.22}$	$0.51_{0.17}$	$0.46_{0.16}$	*0.42*$_{0.18}$	$0.43_{0.17}$
PA	$0.30_{0.14}$	$0.32_{0.10}$	$0.32_{0.12}$	$0.34_{0.13}$	$0.29_{0.13}$	$\mathbf{0.24_{0.11}}$	$0.40_{0.14}$	$0.59_{0.20}$	$0.37_{0.22}$	$0.33_{0.12}$	$0.34_{0.12}$	$0.34_{0.13}$	$0.49_{0.19}$	$0.33_{0.11}$	$0.30_{0.12}$	*0.27*$_{0.12}$
SS	$0.38_{0.14}$	$0.46_{0.16}$	$0.37_{0.12}$	$0.43_{0.15}$	$0.41_{0.12}$	$0.42_{0.13}$	$0.48_{0.18}$	$0.81_{0.25}$	$0.54_{0.16}$	$0.37_{0.15}$	$0.38_{0.13}$	$0.37_{0.15}$	$0.63_{0.15}$	$\mathbf{0.35_{0.15}}$	$0.36_{0.12}$	*0.35*$_{0.12}$
SU	$\mathbf{0.22_{0.13}}$	$0.28_{0.18}$	$0.27_{0.10}$	$0.28_{0.17}$	$0.27_{0.12}$	$0.28_{0.10}$	$0.42_{0.14}$	$0.83_{0.23}$	$0.42_{0.14}$	$0.26_{0.12}$	$0.27_{0.11}$	*0.25*$_{0.13}$	$0.36_{0.16}$	$0.26_{0.12}$	$0.31_{0.11}$	$0.31_{0.12}$
TA	$0.54_{0.10}$	$0.50_{0.09}$	$0.48_{0.07}$	$0.52_{0.11}$	$0.50_{0.08}$	$0.54_{0.11}$	$0.62_{0.12}$	$0.63_{0.12}$	$0.58_{0.13}$	$0.47_{0.06}$	$0.47_{0.07}$	$\mathbf{0.46_{0.07}}$	$0.86_{0.16}$	*0.46*$_{0.06}$	$0.50_{0.09}$	$0.52_{0.09}$
NT	$0.04_{0.03}$	$0.04_{0.02}$	$0.05_{0.03}$	$0.04_{0.02}$	$0.04_{0.03}$	$0.04_{0.02}$	$0.05_{0.02}$	*0.03*$_{0.02}$	$0.03_{0.03}$	$0.03_{0.02}$	$0.03_{0.02}$	$\mathbf{0.02_{0.02}}$	$0.03_{0.02}$	$0.03_{0.02}$	$0.04_{0.03}$	$0.04_{0.03}$
BS	$0.03_{0.01}$	$0.03_{0.01}$	$0.17_{0.03}$	$0.03_{0.01}$	$0.03_{0.02}$	$0.04_{0.01}$	$0.09_{0.03}$	$0.11_{0.02}$	$0.11_{0.19}$	*0.00*$_{0.01}$	$0.00_{0.01}$	$0.16_{0.02}$	$0.03_{0.01}$	$\mathbf{0.00_{0.00}}$	$0.03_{0.02}$	$0.03_{0.02}$
SW	$\mathbf{0.44_{0.04}}$	$0.48_{0.04}$	$0.45_{0.03}$	$0.48_{0.04}$	$0.45_{0.04}$	$0.45_{0.04}$	$0.45_{0.03}$	$0.45_{0.03}$	$0.48_{0.04}$	$0.45_{0.03}$	$0.45_{0.03}$	$0.58_{0.03}$	*0.44*$_{0.03}$	$0.45_{0.03}$	$0.46_{0.04}$	$0.46_{0.04}$
CA	*0.01*$_{0.01}$	$0.01_{0.01}$	$0.03_{0.01}$	$0.02_{0.01}$	$\mathbf{0.00_{0.00}}$	$0.03_{0.01}$	$0.18_{0.01}$	$1.45_{0.55}$	$0.12_{0.03}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.05_{0.01}$	$0.04_{0.01}$	$0.01_{0.00}$	$0.01_{0.01}$	$0.01_{0.01}$
BO	$0.64_{0.10}$	$0.60_{0.10}$	$0.59_{0.10}$	$0.57_{0.11}$	$0.59_{0.12}$	$0.67_{0.16}$	$0.65_{0.17}$	$0.95_{0.32}$	$0.80_{0.21}$	$0.62_{0.09}$	$0.61_{0.09}$	$0.63_{0.08}$	$0.62_{0.06}$	$0.61_{0.08}$	*0.53*$_{0.11}$	$\mathbf{0.53_{0.10}}$
TO	$0.05_{0.02}$	$0.06_{0.03}$	$0.06_{0.03}$	$0.06_{0.03}$	$0.07_{0.03}$	$0.06_{0.03}$	$0.08_{0.03}$	$0.98_{0.04}$	$0.06_{0.03}$	$\mathbf{0.02_{0.01}}$	*0.02*$_{0.01}$	$0.11_{0.03}$	$0.05_{0.02}$	$0.02_{0.01}$	$0.05_{0.02}$	$0.05_{0.03}$
EU	$0.41_{0.04}$	$0.51_{0.04}$	$0.40_{0.03}$	$0.51_{0.04}$	$0.40_{0.03}$	$0.48_{0.04}$	$0.53_{0.05}$	$1.94_{0.25}$	$0.57_{0.08}$	$0.40_{0.04}$	*0.39*$_{0.03}$	$0.40_{0.03}$	$\mathbf{0.33_{0.04}}$	$0.40_{0.04}$	$0.41_{0.04}$	$0.41_{0.04}$
LE	*0.40*$_{0.03}$	$0.41_{0.03}$	$0.41_{0.02}$	$0.41_{0.03}$	$\mathbf{0.40_{0.03}}$	$0.41_{0.03}$	$0.41_{0.03}$	$0.41_{0.03}$	$0.42_{0.03}$	$0.41_{0.02}$	$0.41_{0.02}$	$0.51_{0.04}$	$0.42_{0.03}$	$0.41_{0.02}$	$0.43_{0.03}$	$0.43_{0.03}$
AU	$0.36_{0.10}$	$0.39_{0.10}$	$0.39_{0.08}$	$0.39_{0.10}$	$0.37_{0.08}$	$0.50_{0.09}$	$0.54_{0.09}$	$0.95_{0.69}$	$0.85_{0.15}$	$0.42_{0.09}$	$0.39_{0.08}$	$0.39_{0.08}$	$0.59_{0.13}$	$0.40_{0.09}$	$\mathbf{0.35_{0.08}}$	*0.36*$_{0.08}$
WI	$0.41_{0.02}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.41_{0.02}$	$0.44_{0.02}$	$0.43_{0.02}$	$0.44_{0.02}$	$0.45_{0.03}$	$0.42_{0.02}$	$0.42_{0.02}$	$0.39_{0.02}$	$0.42_{0.02}$	$0.42_{0.02}$	*0.37*$_{0.02}$	$\mathbf{0.36_{0.02}}$
ES	$0.32_{0.04}$	$0.35_{0.04}$	$0.31_{0.04}$	$0.34_{0.03}$	$0.30_{0.04}$	$0.32_{0.04}$	$0.32_{0.03}$	$0.31_{0.04}$	$0.46_{0.63}$	$0.30_{0.04}$	$\mathbf{0.30_{0.03}}$	$0.37_{0.04}$	*0.30*$_{0.03}$	$0.31_{0.04}$	$0.34_{0.03}$	$0.34_{0.03}$
ER	$1.29_{0.07}$	$2.02_{0.12}$	$1.22_{0.05}$	$1.91_{0.15}$	$1.24_{0.04}$	$\mathbf{1.18_{0.02}}$	$1.24_{0.04}$	$1.22_{0.05}$	$1.26_{0.06}$	*1.21*$_{0.06}$	$1.21_{0.04}$	$1.78_{0.10}$	$1.24_{0.05}$	$1.22_{0.04}$	$1.25_{0.04}$	$1.26_{0.05}$
A_{OR}	$0.37_{0.30}$	$0.43_{0.45}$	$0.37_{0.27}$	$0.42_{0.43}$	$0.37_{0.29}$	$0.39_{0.28}$	$0.44_{0.28}$	$0.74_{0.49}$	$0.47_{0.31}$	$0.36_{0.29}$	$0.37_{0.29}$	$0.43_{0.39}$	$0.43_{0.32}$	$\mathbf{0.36_{0.29}}$	$0.36_{0.28}$	*0.36*$_{0.29}$

The best result for each dataset is in bold face and the second one in italics

it was applied to the complete training set and test results were extracted. The criteria for selecting the best configuration were both MAE and MZE performances, depending on the measure we were interested in. The parameter configurations explored are now specified. The Gaussian kernel function was considered for all the kernel methods (SVC1V1, SVC1VA, SVR, CSSVC, SVMOP, SVOREX, SVORIM, REDSVM and KDLOR). The following values were considered for the width of the kernel, $\sigma \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. The cost parameter C of all SVM methods was selected within the values $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and for the KDLOR within the values $C \in \{10^{-1}, 10^{0}, 10^{1}\}$, since, in this case, this parameter presents a different interpretation and, therefore, there is no need to use a larger spectrum of values. An additional parameter u was also needed by KDLOR, which is intended to avoid singularities in the covariance matrices. The values considered were $u \in \{10^{-6}, 10^{-5}, \ldots, 10^{-2}\}$. The range of ϵ for ϵ-SVR was $\epsilon \in \{10^{0}, 10^{1}, \ldots, 10^{3}\}$. For the neural network algorithms (NNOP and NNPOM), the number of hidden neurons, H, was selected by considering the following values, $H \in \{5, 10, 20, 30, 40\}$. The sigmoidal activation function was considered for hidden neurons. For the iRProp+ algorithm, the number of iterations, $iter$, was also decided by cross-validation, by considering the values $iter \in \{50, 100, 150, \ldots, 500\}$. The other parameters of iRProp+ were set as in [129]. For ELMOP, higher numbers of hidden neurons are considered, $H \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, given that it relies on sufficiently informative random projections [72]. With regards to the GPOR algorithm, the hyperparameters are determined by part of the optimisation process. The ORBoost process did not involve the optimisation of any hyperparameter.

Each pair of algorithms is compared by means of the

Wilcoxon test [133]. A level of significance of $\alpha = 0.1$ was considered, and the corresponding correction for the number of comparisons was also included. As 16 algorithms are compared, the total number of comparisons for each dataset is 120, so the corrected level of significance was $\alpha^* = 0.1/120 = 0.00083$.

4.2 Discretised regression datasets

Tables 6 and 7 show the results obtained for all algorithms throughout the discretised regression datasets (when considering $Q = 5$ and $Q = 10$ bins), and also the ordinal regression ones (analysed in the following subsection). The results include the average and the standard deviation of MZE and MAE, respectively. Additionally, the average values for discretised regression problems of 5 classes (A_{D5}), of 10 classes (A_{D10}), for all discretised regression problems (A_D) and for real ordinal regression ones (A_{OR}) are included, as a reference of the mean performance of each method. For all methods including a model selection process, Table 6 shows the results when using MZE as the selection criterion, while the results in Table 7 are obtained using MAE for this selection. In addition, the average computational time including cross-validation, training and test are presented in Table 8. For all these Tables, the best method for each dataset (or set of datasets) is highlighted in bold face and the second one in italics.

As previously stated, the Wilcoxon test was applied to check the existence of significant differences. Using this test, each pair of methods was compared for each dataset and the total number of statistically significant wins or losses was recorded, together with the number of draws (or absence of statistically significant differences). These results are included in Table 9 for the discretised regression datasets. The number of wins (w), draws (d) and losses (l) for $15 \times 24 = 360$ comparisons are included (24 datasets and 15 methods to compare each method against). The methods are ordered by the number of statistically significant wins.

By analysing Tables 6, 7, 8, and specially Table 9, the best performing ordinal regression methods from the different families in the taxonomy can be obtained. From the naïve approaches, SVC1V1 obtains better accuracy or MZE while SVR is better on MAE. However, both improved performances imply worse results for MAE and MZE, respectively. The computational time of SVR is higher, given that an additional parameter has to be cross-validated. In general, SVC1VA and CSSVC show worse MAE and MZE than SVC1V1. When considering ordinal binary decompositions, SVMOP is the best performing one, improving MAE and MZE with respect to NNOP and ELMOP methods. ELMOP is slightly better than NNOP in MAE, but the opposite happens when observing MZE. However, ELMOP is clearly the fastest ordinal binary decomposition method. From the threshold methods, it is clear that GPOR is the best performing one in MZE, REDSVM obtains the highest MAE (although with higher computational cost), and the POM is the fastest one, followed by SVOREX. ORBALL is better in MAE, while ORBLR results in lower MZE and time.

TABLE 8
Average cross-validation, training and test time results for each method and for all datasets.

	Average computational time			
Method	A_{D5}	A_{D10}	A_D	A_{OR}
SCV1V1	$30.4_{13.9}$	$30.2_{15.5}$	$30.3_{14.4}$	$31.3_{29.3}$
SVC1VA	$33.5_{50.1}$	$50.6_{91.5}$	$42.0_{72.7}$	$60.6_{77.6}$
SVR	$91.7_{102.9}$	$96.1_{122.3}$	$93.9_{110.5}$	$225.1_{433.7}$
CSSVC	$37.6_{66.2}$	$58.0_{126.7}$	$47.8_{99.4}$	$63.3_{78.3}$
SVMOP	$67.7_{123.1}$	$125.9_{251.3}$	$96.8_{195.8}$	$130.2_{181.7}$
NNOP	$904.1_{978.8}$	$493.9_{671.0}$	$699.0_{847.0}$	$791.9_{1339.5}$
ELMOP	$2.6_{1.4}$	$3.3_{2.0}$	$2.9_{1.7}$	$3.9_{2.2}$
POM	$\mathbf{0.7_{1.1}}$	$\mathbf{0.8_{1.2}}$	$\mathbf{0.7_{1.1}}$	$\mathbf{1.6_{2.6}}$
NNPOM	$1817.8_{1388.2}$	$3649.2_{3089.5}$	$2733.5_{2522.2}$	$3566.8_{5666.8}$
SVOREX	$91.1_{263.8}$	$137.5_{377.4}$	$114.3_{319.3}$	$410.2_{668.9}$
SVORIM	$159.6_{504.4}$	$307.6_{883.9}$	$233.6_{707.8}$	$524.2_{1132.9}$
KDLOR	$228.5_{441.2}$	$271.4_{522.8}$	$250.0_{473.6}$	$571.1_{889.7}$
GPOR	$2386.6_{6865.7}$	$4250.4_{13087.4}$	$3318.5_{10264.9}$	$27324.2_{80372.2}$
REDSVM	$254.8_{802.3}$	$392.9_{1198.1}$	$323.9_{999.7}$	$935.2_{2477.7}$
ORBALL	$39.1_{41.0}$	$41.2_{40.5}$	$40.1_{39.9}$	$52.8_{65.9}$
ORBLR	$39.2_{41.2}$	$38.7_{41.2}$	$39.0_{40.3}$	$53.0_{66.6}$

The best result is in bold face and the second one in italics

TABLE 9
Wilcoxon tests over discretised regression datasets.

MZE				MAE				Time			
Method	w	d	l	Method	w	d	l	Method	w	d	l
GPOR	211	145	4	**REDSVM**	202	157	1	**POM**	357	0	3
SVOREX	155	199	6	SVORIM	199	158	3	**ELMOP**	318	8	34
SVORIM	139	205	16	GPOR	170	165	25	SVOREX	265	16	79
REDSVM	138	208	14	POM	163	120	77	SVORIM	222	33	105
POM	128	154	78	SVOREX	159	175	26	**SVC1VA**	180	77	103
ORBLR	81	206	73	**SVR**	145	171	44	CSSVC	168	101	91
SVMOP	76	223	61	ORBALL	125	179	56	ORBLR	157	79	124
SVC1V1	68	220	72	ORBLR	120	174	66	SVC1V1	156	55	149
KDLOR	66	225	69	**SVMOP**	95	161	104	ORBALL	146	85	129
ORBALL	60	203	97	KDLOR	73	193	94	REDSVM	135	71	154
SVR	57	220	83	SVC1V1	53	152	155	SVMOP	110	84	166
NNOP	32	200	128	ELMOP	52	139	169	NNOP	93	2	265
CSSVC	28	196	136	NNOP	50	166	144	SVR	75	39	246
SVC1VA	20	204	136	CSSVC	17	114	229	KDLOR	61	54	245
ELMOP	19	181	160	SVC1VA	15	109	236	GPOR	44	76	240
NNPOM	16	183	161	NNPOM	14	123	223	NNPOM	2	2	356

Best method of each family in the taxonomy is highlighted in bold face

4.3 Ordinal regression datasets

This section presents the study performed on the ordinal regression datasets. The objective is to analyse how the methods perform in more realistic situations, where the underlying variable is really unobservable and traditional classification problems appear (e.g. class imbalance). Tables 6, 7 and 8 show the complete set of results obtained and Table 10 includes the corresponding Wilcoxon tests. Its format is similar to that in Table 9 presented in the previous subsection, but the number of comparisons is now $15 \times 17 = 255$. In general, conclusions similar to the ones presented previously can now be obtained, but there are some differences. SVC1V1 performance regarding MZE is now better, when compared to the rest of methods. The problems associated with

TABLE 10
Wilcoxon tests over ordinal regression datasets.

MZE				MAE				Time			
Method	w	d	l	Method	w	d	l	Method	w	d	l
SVOREX	83	161	11	**REDSVM**	87	158	10	**POM**	246	6	3
SVORIM	77	162	16	SVOREX	86	158	11	**ELMOP**	226	1	28
SVMOP	73	166	16	SVORIM	83	163	9	SVOREX	161	9	85
REDSVM	73	167	15	ORBALL	69	153	33	SVORIM	160	10	85
SVC1V1	70	174	11	ORBLR	67	161	27	**SVC1V1**	157	20	78
ORBLR	60	174	21	**SVMOP**	66	177	12	ORBLR	143	24	88
ORBALL	58	169	28	**SVR**	61	162	32	ORBALL	141	24	90
GPOR	55	115	85	SVC1V1	59	173	23	SVC1VA	124	31	100
SVR	47	168	40	GPOR	53	125	77	CSSVC	113	49	93
CSSVC	46	171	38	KDLOR	45	110	100	REDSVM	106	20	129
SVC1VA	42	169	44	NNPOMOP	42	158	55	SVMOP	94	18	143
NNPOMOP	39	161	55	CSSVC	38	167	50	KDLOR	73	15	167
KDLOR	35	122	98	SVC1VA	37	166	52	SVR	62	5	188
ELMOP	23	142	90	POM	23	89	143	NNPOMOP	54	8	193
NNPOM	20	131	104	ELMOP	22	130	103	GPOR	35	28	192
POM	14	98	143	NNPOM	17	120	118	NNPOM	11	0	244

Best method of each family in the taxonomy is highlighted in bold face

these datasets (mainly uneven distribution ratios) are generally better solved by using this kind of decomposition. With this kind of datasets, ELMOP is worse than NNOP for MZE and MAE, although its computational time is the lowest in binary decompositions. The best option from binary decompositions is SVMOP.When analysing threshold methods, our experiments show that the performance of GPOR is much lower in both MZE and MAE. SVM based threshold models are the best performing ones for both measures, SVOREX achieving the best results in MZE and very close to the best performing method, REDSVM, in MAE. Discarding POM and ELMOP, the lowest computationally time is also associated to them. The modelling of $P(\mathbf{x}|y)$ could be one of the causes of the low computational efficiency of the generative model considered in this work (GPOR which yielded the second highest computational time). Again, ORBLR is a bit better than ORBALL in MZE, while ORBALL is better in MAE, although the differences are not large. With respect to REDSVM, its computational cost is high when compared to SVORIM, SVOREX, ORBoost and SVC methods.

4.4 Discussion

This subsection concludes the experimental study with some final remarks about the results obtained. Several ordinal regression methods (see Tables 9 and 10) can be emphasised according to their error (GPOR, SVOREX, SVORIM, REDSVM and SVMOP) or their computational time (POM or ELMOP). However, there are many factors that can influence the choice of the method, and all of them should be considered.

First of all, it is important to highlight that POM is a linear model and, as such, it is very fast to train (with no associated hyperparameters), but its performance is significantly low (except for MZE in discretised regression datasets). This fact is important, given that, excluding the machine learning area, the POM and its variants are the most widely used ordinal regression methods [25], [81], [84], [88].

Regarding the scalability with respect to the number of classes, the average values in Tables 6, 7 and 8 show that MZE and MAE relative performances scale well for discretised datasets with respect to the number of classes. However, for computational time, some methods scale worse with Q such as GPOR, NNOP, SVMOP and NNPOM. The methods which scale better are SCV1V1, SCV1VA, SVR, CSSVC, KDLOR and ORBoost.

When dealing with large datasets, we conclude that POM is a good option, given the low computational cost needed. The results achieved for MZE and MAE are worse than those of other alternatives, but they are good enough when computational time is a priority.

Our study shows that the naïve approaches can obtain competitive performance and be difficult to beat for some datasets. SVC1V1 achieves very good MZE results for real ordinal regression datasets. However, SVM threshold models improves MAE and MZE results, as well as being simpler models. Indeed, all threshold models allow the visualisation of predicted projections together with the thresholds. This can be used for various purposes, from ranking patterns to trying to discover uncertain predictions (projections very close to class thresholds). This kind of analysis is generally more difficult with nominal models, such as SVC1V1. In general, the SVC1VA alternative has been shown to achieve worse results than SVC1V1 for the three measures evaluated (as previously shown in other studies [59]). CSSVC results are a bit better than those of SVC1VA, but still far from SVC1V1.

Binary decomposition approaches are shown to be good alternatives, especially SVMOP. However, as discussed in Subsection 3.2, their theoretical analysis is more difficult, and it is necessary to decide how to combine different binary predictions.

Of all the threshold models analysed, SVOREX and SVORIM are the best. The computational time required by SVOREX is slightly lower, and it always achieves better results than SVORIM, except for MAE in discretised regression datasets. ORBoost methods show a worse performance than SVOR methods, but they scale better with Q. REDSVM is shown to be quite competitive, but with a higher computational cost.

Neural networks (NNPOM and NNOP) are generally beaten by their SVM counterparts, both in MZE and MAE. Moreover, the training time for these methods and GPOR is generally the highest.

When comparing discretised regression datasets and real ordinal regression ones, some performance differences can be highlighted. For example, GPOR performance is seriously affected when dealing with real ordinal classification datasets. In general, SVM methods are more robust in the derived problems that can appear with these datasets. This is an important point in our study, because many of the ordinal regression works in the literature make use of discretised regression sets,

hiding some possible difficulties of the methods when dealing with problems such as imbalanced distributions.

When real ordinal regression datasets are considered, POM and the GPOR performances decrease (both in MZE and MAE) drastically. Both models have one feature in common. They assume that their perturbation terms follow certain distribution functions (a logistic distribution in the case of the POM model or a Gaussian distribution in the case of the GPOR model). These distributional assumptions perform correctly in discretised regression datasets, but not for real ordinal regression datasets.

5 Conclusions

This paper offers an exhaustive survey of the ordinal regression methods proposed in the literature. The problem setting has been clearly established and differentiated from other ranking topics. After this, a taxonomy of ordinal regression methods is proposed, dividing them into three main groups: naïve approaches, binary decompositions and threshold models. Furthermore, the most important methods of each family (a total of 16 methods) are empirically evaluated in two kinds of datasets, 24 discretised regression datasets and 17 real ordinal regression ones.

The taxonomy proposed can help the researcher or the practitioner choose the best method for a concrete problem, considering also the empirical results herein provided. It can also assist researchers in developing and proposing new methods, providing a way to classify them and to select the most similar ones. The results presented in this paper confirm that there is no single method which performs the best in all possible datasets and problem requirements. However, these results can be used to discard some of the methods, especially those clearly presenting worse performance or too high computational time. We would like to stress certain methods: 1) SVC1V1 as representative of the naïve approaches, achieving an especially good MZE because of the recursive partitioning of all pairs of classes; 2) SVMOP achieves the best results from ordinal binary decomposition methods; 3) ELMOP or POM are a good option if the computational cost is a priority; and 4) SVOREX and SVORIM can be considered as the best threshold models, showing competitive accuracy, MAE and time values. Finally, there is a website (http://www.uco.es/grupos/ayrna/orreview) collecting the implementations of the methods in this survey, the detailed results, the datasets and the corresponding statistical analysis.

References

[1] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[2] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Data Management Systems. Morgan Kaufmann (Elsevier), 2005.

[3] V. Cherkassky and F. M. Mulier, *Learning from Data: Concepts, Theory, and Methods*. Wiley-Interscience, 2007.

[4] J. A. Anderson, "Regression and ordered categorical variables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 1, pp. 1–30, 1984.

[5] R. Bender and U. Grouven, "Ordinal logistic regression in medical research," *J. R. Coll. Physicians Lond.*, vol. 31, no. 5, pp. 546–551, 1997.

[6] ——, "Using binary logistic regression models for ordinal data with non-proportional odds," *J. Clin. Epidemiol.*, vol. 51, no. 10, pp. 809–816, 1998.

[7] W. M. Jang, S. J. Eun, C.-E. Lee, and Y. Kim, "Effect of repeated public releases on cesarean section rates," *J. Prev. Med. Pub. Health*, vol. 44, no. 1, pp. 2–8, 2011.

[8] O. M. Doyle, E. Westman, A. F. Marquand, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, S. C. Williams *et al.*, "Predicting progression of alzheimer's disease using ordinal regression," *PloS one*, vol. 9, no. 8, p. e105542, 2014.

[9] M. Pérez-Ortiz, P. A. Gutiérrez, C. García-Alonso, L. Salvador-Carulla, J. A. Salinas-Pérez, and C. Hervás-Martínez, "Ordinal classification of depression spatial hot-spots of prevalence," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA)*, nov. 2011, pp. 1170 –1175.

[10] J. S. Cardoso, J. F. P. da Costa, and M. Cardoso, "Modelling ordinal relations with SVMs: an application to objective aesthetic evaluation of breast cancer conservative treatment," *Neural Networks*, vol. 18, no. 5-6, pp. 808–817, 2005.

[11] M. Pérez-Ortiz, M. Cruz-Ramírez, M. Ayllón-Terán, N. Heaton, R. Ciria, and C. Hervás-Martínez, "An organ allocation system for liver transplantation based on ordinal regression," *Applied Soft Computing Journal*, vol. 14, pp. 88–98, 2014.

[12] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung, "Ordinal hyperplanes ranker with cost sensitivities for age estimation," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2011, pp. 585 –592.

[13] J. W. Yoon, S. J. Roberts, M. Dyson, and J. Q. Gan, "Bayesian inference for an adaptive ordered probit model: An application to brain computer interfacing," *Neural Networks*, vol. 24, no. 7, pp. 726–734, 2011.

[14] Y. Kwon, I. Han, and K. Lee, "Ordinal pairwise partitioning (opp) approach to neural networks training in bond rating," *Intelligent Systems in Accounting Finance and Management*, vol. 6, no. 1, pp. 23–40, 1997.

[15] K.-j. Kim and H. Ahn, "A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach," *Computers and Operations Research*, vol. 39, no. 8, pp. 1800–1811, 2012.

[16] H. Dikkers and L. Rothkrantz, "Support vector machines in ordinal classification: An application to corporate credit scoring," *Neural Network World*, vol. 15, no. 6, pp. 491–507, 2005.

[17] F. Fernández-Navarro, P. Campoy-Muñoz, M.-D. La Paz-Marín, C. Hervás-Martínez, and X. Yao, "Addressing the EU sovereign ratings using an ordinal regression approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2228–2240, 2013.

[18] M. J. Mathieson, "Ordinal models for neural networks," in *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, ser. Neural Networks in Financial Engineering, J. M. A.-P. N. Refenes, Y. Abu-Mostafa and A. Weigend, Eds. World Scientific, 1996, pp. 523–536.

[19] M. Kim and V. Pavlovic, "Structured output ordinal regression for dynamic facial emotion intensity prediction," in *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010), Part III*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6313. Springer Berlin Heidelberg, 2010, pp. 649–662.

[20] O. Rudovic, V. Pavlovic, and M. Pantic, "Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2634–2641.

[21] ——, "Kernel conditional ordinal random fields for temporal segmentation of facial action units," in *Proceedings of the European Conference on Computer Vision Computer Vision (ECCV 2012). Part II*, ser. Lecture Notes in Computer Science, A. Fusiello, V. Murino, and R. Cucchiara, Eds., vol. 7584. Springer Berlin Heidelberg, 2012, pp. 260–269.

[22] H. Yan, "Cost-sensitive ordinal regression for fully automatic facial beauty assessment," *Neurocomputing*, vol. 129, pp. 334–342, 2014.

[23] Q. Tian, S. Chen, and X. Tan, "Comparative study among three strategies of incorporating spatial structures to ordinal image regression," *Neurocomputing*, vol. 136, no. 0, pp. 152 – 161, 2014.

[24] P. A. Gutiérrez, S. Salcedo-Sanz, C. Hervás-Martínez, L. Carro-Calvo, J. Sánchez-Monedero, and L. Prieto, "Ordinal and nominal classification of wind speed from synoptic pressure patterns," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 3, pp. 1008–1015, 2013.

[25] A. S. Fullerton and J. Xu, "The proportional odds with partial proportionality constraints model for ordinal response variables," *Soc. Sci. Res.*, vol. 41, no. 1, pp. 182–198, 2012.

[26] S. Baccianella, A. Esuli, and F. Sebastiani, "Feature selection for ordinal text classification," *Neural Comput.*, vol. 26, no. 3, pp. 557–591, 2014.

[27] C.-W. Seah, I. W. Tsang, and Y.-S. Ong, "Transductive ordinal regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1074–1086, 2012.

[28] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 115–132.

[29] W. Chu and S. S. Keerthi, "Support Vector Ordinal Regression," *Neural Comput.*, vol. 19, no. 3, pp. 792–815, 2007.

[30] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *J. of Machine Learning Research*, vol. 6, pp. 1019–1041, 2005.

[31] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li, "Kernel discriminant learning for ordinal regression," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 906–910, 2010.

[32] J. C. Hühn and E. Hüllermeier, "Is an ordinal class structure useful in classifier learning?" *International Journal of Data Mining, Modelling and Management*, vol. 1, no. 1, pp. 45–67, 2008.

[33] A. Ben-David, L. Sterling, and T. Tran, "Adding monotonicity to learning algorithms may impair their accuracy," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6627–6634, 2009.

[34] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[35] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez, "An experimental study of different ordinal regression methods and measures," in *7th International Conference on Hybrid Artificial Intelligence Systems*, 2012, pp. 296–307.

[36] S. Rajaram, A. Garg, X. S. Zhou, and T. S. Huang, "Classification approach towards ranking and sorting problems," in *Proc. of the 14th European Conference on Machine Learning (ECML)*, ser. Lecture Notes in Computer Science, vol. 2837, 2003, pp. 301–312.

[37] S. Rajaram and S. Agarwal, "Generalization bounds for k-partite ranking," in *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2005)*, 2005, pp. 28–23.

[38] J. Fürnkranz, E. Hüllermeier, and S. Vanderlooy, "Binary decomposition methods for multipartite ranking," in *Proc. of the European Conference on Machine Learning (ECML)*, ser. Lecture Notes in Computer Science, vol. 578, no. I, 2009, pp. 359–374.

[39] K. Uematsu and Y. Lee, "Statistical optimality in multipartite ranking and ordinal regression," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. In Press, 2015.

[40] T.-Y. Liu, *Learning to Rank for Information Retrieval*. Springer-Verlag, 2011.

[41] H.-T. Lin and L. Li, "Reduction from cost-sensitive ordinal ranking to weighted binary classification," *Neural Comput.*, vol. 24, no. 5, pp. 1329–1367, 2012.

[42] Q. Hu, W. Pan, L. Zhang, D. Zhang, Y. Song, M. Guo, and D. Yu, "Feature selection for monotonic classification," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 69–81, 2012.

[43] Q. Hu, X. Che, L. Zhang, D. Zhang, M. Guo, and D. Yu, "Rank entropy-based decision trees for monotonic classification," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2052 –2064, 2012.

[44] W. Kotłowski, K. Dembczyński, S. Greco, and R. Słowiński, "Stochastic dominance-based rough set model for ordinal classification," *Inf. Sciences*, vol. 178, no. 21, pp. 4019–4037, 2008.

[45] W. Kotlowski and R. Slowiński, "On nonparametric ordinal classification with monotonicity constraints," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2576–2589, 2013.

[46] R. Potharst and A. J. Feelders, "Classification trees for problems with monotonicity constraints," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 1–10, jun 2002.

[47] C.-W. Seah, I. Tsang, and Y.-S. Ong, "Transfer ordinal label learning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1863–1876, 2013.

[48] J. Alonso, J. J. Coz, J. Díez, O. Luaces, and A. Bahamonde, "Learning to predict one or more ranks in ordinal regression tasks," in *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML), Part I*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds., vol. 5211. Springer Berlin Heidelberg, 2008, pp. 39–54.

[49] J. Verwaeren, W. Waegeman, and B. De Baets, "Learning partial ordinal class memberships with kernel-based proportional odds models," *Computational Statistics & Data Analysis*, vol. 56, no. 4, pp. 928–942, 2012.

[50] D. Devlaminck, W. Waegeman, B. Bauwens, B. Wyns, P. Santens, and G. Otte, "From circular ordinal regression to multilabel classification," in *Proceedings of the 2010 Workshop on Preference Learning (European Conference on Machine Learning, ECML)*, 2010.

[51] V. Torra, J. Domingo-Ferrer, J. M. Mateo-Sanz, and M. Ng, "Regression for ordinal variables without underlying continuous variables," *Information Sciences*, vol. 176, no. 4, pp. 465–474, 2006.

[52] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[53] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, August 2007.

[54] S. Kramer, G. Widmer, B. Pfahringer, and M. D. Groeve, "Prediction of ordinal classes using regression trees," *Fundamenta Informaticae*, vol. 47, no. 1-2, pp. 1–13, 2001.

[55] E. F. Harrington, "Online ranking/collaborative filtering using the perceptron algorithm," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML2003)*, 2003.

[56] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Exploitation of pairwise class distances for ordinal classification." *Neural Comput.*, vol. 25, no. 9, pp. 2450–2485, 2013.

[57] A. Agresti, *Analysis of ordinal categorical data*, ser. Wiley Series in Probability and Statistics. Wiley, 2010.

[58] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications*, vol. 16, pp. 264–280, 1971.

[59] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[60] H.-H. Tu and H.-T. Lin, "One-sided support vector regression for multiclass cost-sensitive classification," in *Proceedings of the Twenty-Seventh International Conference on Machine learning (ICML2010)*, 2010, pp. 49–56.

[61] S. B. Kotsiantis and P. E. Pintelas, "A cost sensitive technique for ordinal classification problems," in *Methods and applications of artificial intelligence (Proc. of the 3rd Hellenic Conference on Artificial Intelligence, SETN)*, ser. Lecture Notes in Artificial Intelligence, vol. 3025, 2004, pp. 220–229.

[62] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *J. of Machine Learning Research*, vol. 1, pp. 113–141, Sep. 2001.

[63] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proceedings of the 12th European Conference on Machine Learning*, ser. EMCL '01. London, UK: Springer-Verlag, 2001, pp. 145–156.

[64] J. S. Cardoso and J. F. P. da Costa, "Learning to classify ordinal data: The data replication method," *J. of Machine Learning Research*, vol. 8, pp. 1393–1429, 2007.

[65] W. Waegeman and L. Boullart, "An ensemble of weighted support vector machines for ordinal regression," *International Journal of Computer Systems Science and Engineering*, vol. 3, no. 1, pp. 47–51, 2009.

[66] H. Wu, H. Lu, and S. Ma, "A practical SVM-based algorithm for ordinal regression in image retrieval," in *Proceedings of the eleventh ACM international conference on Multimedia (Multimedia2003)*, 2003, pp. 612–621.

[67] M. Pérez-Ortiz, P. A. Gutiérrez, and C. Hervás-Martínez, "Projection based ensemble learning for ordinal regression," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 681–694, 2014.

[68] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer, August 2001.

[69] M. Costa, "Probabilistic interpretation of feedforward network outputs, with relationships to statistical prediction of ordinal quantities," *Int. J. Neural Syst.*, vol. 7, no. 5, pp. 627–638, 1996.

[70] K. Crammer and Y. Singer, "Pranking with ranking," in *Advances in Neural Information Processing Systems*, vol. 14. MIT Press, 2001, pp. 641–647.

[71] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2008, IEEE World Congress on Computational Intelligence)*. IEEE Press, 2008, pp. 1279–1284.

[72] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 513–529, 2012.

[73] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, and X. Wang, "Ordinal extreme learning machine," *Neurocomputing*, vol. 74, no. 1–3, pp. 447–456, 2010.

[74] J. Sánchez-Monedero, P. A. Gutiérrez, and C. Hervás-Martínez, "Evolutionary ordinal extreme learning machine," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8073 LNAI, pp. 500–509, 2013.

[75] F. Fernandez-Navarro, A. Riccardi, and S. Carloni, "Ordinal neural networks without iterative tuning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2075–2085, 2014.

[76] J. F. P. da Costa and J. Cardoso, "Classification of ordinal data using neural networks," in *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, ser. Lecture Notes in Computer Science, J. Gama, R. Camacho, P. Brazdil, A. Jorge, and L. Torgo, Eds., vol. 3720. Springer Berlin Heidelberg, 2005, pp. 690–697.

[77] J. F. P. da Costa, H. Alonso, and J. S. Cardoso, "The unimodal model for the classification of ordinal data," *Neural Networks*, vol. 21, pp. 78–91, January 2008.

[78] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[79] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 737–744.

[80] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 42, no. 2, pp. 109–142, 1980.

[81] A. Agresti, *Categorical Data Analysis*, 2nd ed. John Wiley and Sons, 2002.

[82] M. J. Mathieson, "Ordered classes and incomplete examples in classification," in *Proceedings of the 1996 Conference on Neural Information Processing Systems (NIPS)*, ser. Advances in Neural Information Processing Systems, T. P. Michael C. Mozer, Michael I. Jordan, Ed., vol. 9, 1999, pp. 550–556.

[83] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed., ser. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 1989.

[84] R. Williams, "Generalized ordered logit/partial proportional odds models for ordinal dependent variables," *Stata Journal*, vol. 6, no. 1, pp. 58–82, March 2006.

[85] B. Peterson and J. Harrell, Frank E., "Partial proportional odds models for ordinal response variables," *Journal of the Royal Statistical Society*, vol. 39, no. 2, pp. 205–217, 1990, series C.

[86] G. Tutz, "Generalized semiparametrically structured ordinal models," *Biometrics*, vol. 59, no. 2, pp. 263–273, 2003.

[87] M. Dorado-Moreno, P. A. Gutiérrez, and C. Hervás-Martínez, "Ordinal classification using hybrid artificial neural networks with projection and kernel basis functions," in *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS2012)*, 2012, p. 319–330.

[88] T. Van Gestel, B. Baesens, P. Van Dijcke, J. Garcia, J. Suykens, and J. Vanthienen, "A process model to develop an internal rating system: Sovereign credit ratings," *Decision Support Systems*, vol. 42, no. 2, pp. 1131–1151, 2006.

[89] J. D. Rennie and N. Srebro, "Loss functions for preference levels: Regression with discrete ordered labels," in *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*. Kluwer Norwell, MA, 2005, pp. 180–186.

[90] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN 99)*, vol. 1, 1999, pp. 97–102.

[91] A. Shashua and A. Levin, "Ranking with large margin principle: two approaches," in *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2003)*, ser. Advances in Neural Information Processing Systems, no. 16. MIT Press, 2003, pp. 937–944.

[92] W. Chu and S. S. Keerthi, "New approaches to support vector ordinal regression," in *In ICML'05: Proceedings of the 22nd international conference on Machine Learning*, 2005, pp. 145–152.

[93] S. Agarwal, "Generalization bounds for some ordinal regression algorithms," in *Algorithmic Learning Theory*, ser. Lecture Notes in Artificial Intelligence (Lecture Notes in Computer Science). Springer-Verlag Berlin Heidelberg, 2008, vol. 5254, pp. 7–21.

[94] E. Carrizosa and B. Martin-Barragan, "Maximizing upgrading and downgrading margins for ordinal regression," *Mathematical Methods of Operations Research*, vol. 74, no. 3, pp. 381–407, 2011.

[95] B. Zhao, F. Wang, and C. Zhang, "Block-quantized support vector ordinal regression," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 882–890, 2009.

[96] B. Gu, J.-D. Wang, and T. Li, "Ordinal-class core vector machine," *J. of Comp. Science and Technology*, vol. 25, no. 4, pp. 699–708, 2010.

[97] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *J. of Machine Learning Research*, vol. 6, pp. 363–392, 2005.

[98] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. In Press, 2015.

[99] J. S. Cardoso, R. Sousa, and I. Domingues, "Ordinal data classification using kernel discriminant analysis: A comparison of three approaches," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 1, 2012, pp. 473–477.

[100] B.-Y. Sun, H.-L. Wang, W.-B. Li, H.-J. Wang, J. Li, and Z.-Q. Du, "Constructing and combining orthogonal projection vectors for ordinal regression," *Neural Processing Letters*, pp. 1–17, 2014.

[101] Y. Liu, Y. Liu, S. Zhong, and K. C. Chan, "Semi-supervised manifold ordinal regression for image ranking," in *Proceedings of the 19th ACM international conference on Multimedia (ACM MM2011)*. New York, NY, USA: ACM, 2011, pp. 1393–1396.

[102] Y. Liu, Y. Liu, and K. C. C. Chan, "Ordinal regression via manifold learning," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, W. Burgard and D. Roth, Eds. AAAI Press, 2011, pp. 398–403.

[103] Y. Liu, Y. Liu, K. C. C. Chan, and J. Zhang, "Neighborhood preserving ordinal regression," in *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service (ICIMCS12)*. New York, NY, USA: ACM, 2012, pp. 119–122.

[104] K. Crammer and Y. Singer, "Online ranking by projecting," *Neural Comput.*, vol. 17, no. 1, pp. 145–175, 2005.

[105] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Advances in Neural Information Processing Systems*, no. 19, 2007, pp. 865–872.

[106] H.-T. Lin and L. Li, "Combining ordinal preferences by boosting," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2009, pp. 69–83.

[107] ——, "Large-margin thresholded ensembles for ordinal regression: Theory and practice," in *Proc. of the 17th Algorithmic Learning Theory International Conference*, ser. Lecture Notes in Artificial Intelligence (LNAI), J. L. Balcázar, P. M. Long, and F. Stephan, Eds., vol. 4264. Springer-Verlag, October 2006, pp. 319–333.

[108] F. Xia, L. Zhou, Y. Yang, and W. Zhang, "Ordinal regression as multiclass classification," *International Journal of Intelligent Control and Systems*, vol. 12, no. 3, pp. 230–236, 2007.

[109] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *J. of Machine Learning Research*, vol. 4, pp. 933–969, 2003.

[110] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[111] K. Dembczyński, W. Kotłowski, and R. Słowiński, "Ordinal classification with decision rules," in *Mining Complex Data*, ser. Lecture Notes in Computer Science, vol. 4944. Warsaw, Poland: Springer, 2008, pp. 169–181.

[112] A. Riccardi, F. Fernandez-Navarro, and S. Carloni, "Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 44, no. 10, pp. 1898–1909, 2014.

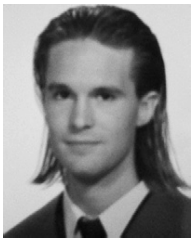
[113] F. Fernández-Navarro, P. A. Gutiérrez, C. Hervás-Martínez, and X. Yao, "Negative correlation ensemble learning for ordinal

regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1836–1849, 2013.
[114] W. Chu and Z. Ghahramani, "Preference learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning (ICML2005)*, 2005, pp. 137–144.
[115] X. Chang, Q. Zheng, and P. Lin, "Ordinal regression with sparse bayesian," in *Proceedings of the 5th International Conference on Intelligent Computing (ICIC2009)*, ser. Lecture Notes in Computer Science, vol. 5755, 2009, pp. 591–599.
[116] P. Srijith, S. Shevade, and S. Sundararajan, "A probabilistic least squares approach to ordinal regression," *Lecture Notes in Artificial Intelligence*, vol. 7691, pp. 683–694, 2012.
[117] ——, "Validation based sparse gaussian processes for ordinal regression," *Lecture Notes in Computer Science*, vol. 7664, no. 2, pp. 409–416, 2012.
[118] ——, "Semi-supervised gaussian process ordinal regression," *Lecture Notes in Artificial Intelligence*, vol. 8190, no. 3, pp. 144–159, 2013.
[119] J. F. Pinto da Costa, R. Sousa, and J. S. Cardoso, "An all-at-once unimodal SVM approach for ordinal classification," in *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA2010)*. IEEE Computer Society Press, 2010, pp. 59–64.
[120] J. Cardoso and R. Sousa, "Classification models with global constraints for ordinal data," in *Proceedings of the Ninth International Conference on Machine Learning and Applications (ICMLA2010)*, 2010, pp. 71 –77.
[121] R. Sousa and J. Cardoso, "Ensemble of decision trees with global constraints for ordinal classification," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA2011)*, 2011, pp. 1164–1169.
[122] S. Fouad and P. Tiňo, "Adaptive metric learning vector quantization for ordinal classification," *Neural Comput.*, vol. 24, pp. 2825–2851, 2012.
[123] S. Fouad and P. Tino, "Prototype based modelling for ordinal classification," in *Proceedings of the 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL2012)*, ser. Lecture Notes in Computer Science, H. Yin, J. Costa, and G. Barreto, Eds., vol. 7435. Springer Berlin Heidelberg, 2012, pp. 208–215.
[124] K. Dembczyański1 and W. Kotlowski, "Decision rule-based algorithm for ordinal classification based on rank loss minimization," in *Proceedings of the 2009 Workshop on Preference Learning (European Conference on Machine Learning, ECML)*, 2009.
[125] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[126] PASCAL, "Pascal (Pattern Analysis, Statistical Modelling and Computational Learning) machine learning benchmarks repository," 2011. [Online]. Available: http://mldata.org/
[127] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, May 2011.
[128] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. of Machine Learning Research*, vol. 5, pp. 975—-1005, 2004.
[129] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, no. 6, pp. 105–123, 2003.
[130] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, 2014.
[131] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA'09)*, 2009, pp. 283–287.
[132] L. Prechelt, "PROBEN1: A set of neural network benchmark problems and benchmarking rules," Fakultät für Informatik (Universität Karlsruhe), Tech. Rep. 21/94, 1994.
[133] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

Pedro Antonio Gutiérrez was born in Córdoba, Spain. He received the B.S. degree in Computer Science from the University of Sevilla, Spain, in 2006, and the Ph.D. degree in Computer Science and Artificial Intelligence from the University of Granada, Spain, in 2009. He is currently an Assistant Professor in the Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. His current research interests include patter recognition, evolutionary computation, and their applications.

María Pérez-Ortiz was born in Córdoba, Spain, in 1990. She received the B.S. degree in computer science in 2011 and the M.Sc. degree in intelligent systems in 2013 from the University of Córdoba, Spain, where she is currently pursuing the Ph.D. degree in computer science and artificial intelligence in the Department of Computer Science and Numerical Analysis. Her current interests include a wide range of topics concerning machine learning and pattern recognition.

Javier Sánchez-Monedero was born in Córdoba (Spain). He received the B.S in Computer Science from the University of Granada, Spain, in 2008 and the M.S. in Multimedia Systems from the University of Granada in 2009, where he obtained the Ph.D. degree on Information and Communication Technologies in 2013. He is working as researcher with the Department of Computer Science and Numerical Analysis at the University of Córdoba. His current research interests include computational intelligence and distributed systems.



Francisco Fernández-Navarro (M'13) was born in Córdoba, Spain, in 1984. He received the M.Sc. degree in computer science from the University of Córdoba, Spain, in 2008, the M.Sc. degree in artificial intelligence from the University of Málaga, Spain, in 2009 and the Ph.D. degree in computer science and artificial intelligence from the University of Málaga, in 2011. He is currently a Research Fellow in computational management with the European Space Agency, Noordwijk, The Netherlands. His current research interests include neural networks, ordinal regression, imbalanced classification and hybrid algorithms.

César Hervás-Martínez was born in Cuenca, Spain. He received the B.S. degree in Statistics and Operations Research from the "Universidad Complutense", Madrid, Spain, in 1978, and the Ph.D. degree in Mathematics from the University of Seville, Spain, in 1986. He is currently a Professor of Computer Science and Artificial Intelligence in the Department of Computer Science and Numerical Analysis, University of Córdoba, and an Associate Professor in the Department of Quantitative Methods, School of Economics, University of Córdoba. His current research interests include neural networks, evolutionary computation, and the modelling of natural systems.

2.2. On the use of nominal and ordinal classifiers for the discrimination of states of development in fish oocytes

This paper considers the problem of tackling the discrimination of development states in fish occytes by an ordinal regression approach. In this sense, the analysis of microscopic images of fish gonad cells (also called oocytes) is a useful tool to estimate parameters of fish reproductive ecology and to analyse fish population dynamics. The best method to classify oocytes is histology, although experienced personnel and a lot of effort is required. To solve this, a software tool has been developed [46] (Govocitos[1], an automatic image analysis software which uses colour texture classification to discriminate oocytes in three main states of development). Oocytes go through different developmental states in a continuum temporal sequence. However, the current oocyte analysis software does not consider this factor, although it is necessary to minimise some kind of errors associated to the ordered nature of this variable.

As said, in the previous approach, the temporal evolution of states was not considered, which limits its usefulness for understanding the oocyte development. In this paper we solve this deficiency considering the whole time line of developmental states, using ordinal classifiers to fully capture the temporal evolution of states and extending the study to species with more states. In this sense, three different fish species are considered to perform the study, two of them presenting three ordinal states of development, and the remaining one with six states of development.

11 ordinal and 13 nominal classifiers have been used with this purpose in the experimentation. Moreover, we also conducted two different experiments to test for the best way or partitioning the data: a leave-one-image-out (LOIO) and a mixing procedure. It is clear that a LOIO approach is more realistic, because no information about the image that is considered for testing is included for training. However, the results for LOIO are worst in this case. This could be due to some factors in the image, such as the lighting conditions or the instrumentation used. These factors could be studied in future research. On the whole, the experiments demonstrate that ordinal classifiers exhibit improved robustness and performance compared to nominal methods for all the species considered. Moreover, the difference between ordinal and nominal techniques has been shown to be higher when the number of states increases, being clearly reflected by ordinal quality measures. Finally, the confusion matrix of the best method (a binary ordinal decomposition method [113]) shows that ordinal classifiers locate their errors in states near to the true ones, with sensitivities and positive predictions above 60 % for almost all the states.

[1] `https://forxa.mancomun.org/projects/govocitos`

Neural Processing Letters manuscript No.
(will be inserted by the editor)

On the use of nominal and ordinal classifiers for the discrimination of states of development in fish oocytes

M. Pérez-Ortiz · M. Fernández-Delgado · E. Cernadas · R. Domínguez-Petit · P.A. Gutiérrez · C. Hervás

Received: date / Accepted: date

Abstract The analysis of microscopic images of fish gonad cells (*oocytes*) is a useful tool to estimate parameters of fish reproductive ecology and to analyze fish population dynamics. The study of oocyte dynamics is needed to understand ovary development and reproductive cycle of fish. Oocytes go through different developmental states in a continuum temporal sequence, not exploited by the current oocyte analysis software, that provides an interesting example of ordinal classification. In this paper we compare 11 ordinal and 13 nominal state-of-the-art classifiers using oocytes of three fish species (*Merluccius merluccius*, *Trisopterus luscus* and *Reinhardtius hippoglossoides*). The best results are achieved by SVMOD, an ordinal decomposition method of the labelling space based on the Support Vector Machine, varying strongly with the number of states for each specie (about 95% and 80% of accuracy with three and six states respectively). The classifiers designed specially for ordinal classification are able to capture the underlying nature of the state ordering much better than common nominal classifiers. This is demonstrated by several metrics specially designed to measure misclassification errors associated to states far in the ranking scale.

We acknowledge support from the "Junta de Andalucía" under project P11-TIC-7508, from the Spanish Ministry of Science and Innovation (MICINN) under projects TIN2011-22935, TIN2012-32262 and TIN2011-22794, and from FEDER funds.

M. Pérez-Ortiz · P.A. Gutiérrez · C. Hervás
Dept. of Computer Science and Numerical Analysis, Univ. of Córdoba, Córdoba, Spain

M. Fernández-Delgado · E. Cernadas
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC, Univ. of Santiago de Compostela, Campus Vida, Tel.: +34-881816458, Fax: +34-881816405
E-mail: manuel.fernandez.delgado@usc.es

R. Domínguez-Petit
Dept. of Fisheries Ecology, Instituto de Investigaciones Marinas, Agencia Estatal Consejo Superior de Investigaciones Científicas, Vigo, Spain

Keywords Fish oocytes · Ordinal classification · Texture analysis · *Reinhardtius hippoglossoides* · Decomposition methods

1 Introduction

The assessment of oocyte development dynamic and fecundity is a fundamental topic in the study of reproductive biology and population dynamics [16]. To estimate fecundity with accuracy, only mature oocytes must be considered, which requires a reliable classification of oocytes according to its state of development. The best method to classify oocytes is histology, although experienced personnel is required. The main developmental states of oocytes are: *Primary Growth* (PG), *Cortical Alveoli* (CA), *Vitellogenic* (VIT), *Hydrated* (HYD) and *Atretic* (AT). The PG state corresponds to immature oocytes; CA, VIT and HYD to mature ones; and AT corresponds to those mature oocytes that will be resorpted (i.e., non-ovulated). Depending on the objective of the study, these main states could be divided in sub-states. Specifically, the specie *Reinhardtius hippoglossoides*, also known as Greenland halibut, presents some irregularities in the maturation processes [23, **?**] that could suggest that individual spawning does not necessarily occur on an annual basis as for most exploited fish. This specie presents a unique reproductive development pattern, with ovaries simultaneously containing oocytes developing for the current and subsequent reproductive seasons [26, 18]. Four sub-levels of development within the VIT state have been identified (VIT1, VIT2, VIT3 and VIT4) in this specie (see Fig. 1). When maturation begins, a group of oocytes evolves from PG to CA and progresses until reach VIT2; then some oocytes (called the leading cohort) continue the progression (VIT3-VIT4-HYD), while the rest of mature oocytes (secondary cohort) remains in VIT2 (likely until the next spawning season) or become AT. To analyze oocyte cohort dynamic and estimate egg production it is necessary to classify correctly the VIT sub-states.

In a previous work [12] we developed Govocitos[1], an automatic image analysis software which uses color texture classification to discriminate oocytes in the four main states of development (CA, VIT, HYD and AT), although states VIT and AT could not be reliably distinguished. Govocitos achieves acceptable accuracies with oocytes of two gadiform species, *Merluccius merluccius* and *Trisopterus luscus*, but it does not consider the VIT sub-levels nor the temporal evolution of states, which limits its usefulness for understanding the oocyte development. In this paper we solve this defficiency considering the whole time line of developmental states, using ordinal classifiers to fully capture the temporal evolution of states and extending the study to species with more states, such as *Reinhardtius hippoglossoides*. Section 2 introduces the ordinal classification setting and describes the most commonly used ordinal classifiers. Section 3 presents the experiments (data acquisition, experimental methodology, tested methods and quality measures) and discusses the results. Finally, Section 4 compiles some conclusions of the work.

[1] `https://forxa.mancomun.org/projects/govocitos`

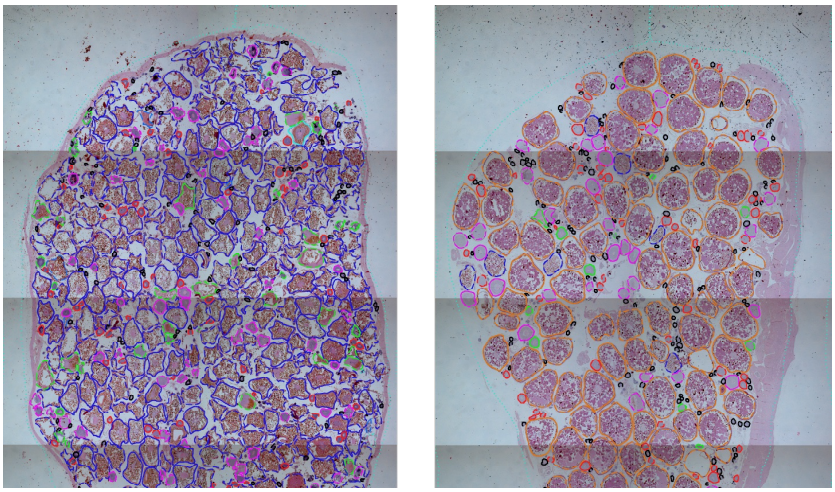


Fig. 1 Examples of histological images of fish specie *Reinhardtius hippoglossoides*. The cell outlines were manually annotated by experts using the Govocitos software tool. The color identifies the state of development of the oocyte: black (PG), red (CA), pink (VIT1), cyan (VIT2), blue (VIT3), orange and green (VIT4).

2 Ordinal classification methods

Ordinal classification is receiving much attention from the pattern recognition and machine learning communities, given its applicability to real world problems (economy, medicine, psychology and others). This paradigm assumes that a natural ordering of the class labels is given, which is an assumption not considered by standard multinomial (or nominal) classifiers or by the common zero-one loss function. In contrast to regression, the categories are finite and the labelling space is non-metric, with unknown distances between categories. In the current paper, the classes correspond to the states of development of oocytes, which are naturally ordered by its growing along the time. This natural order requires to penalize differently the misclassification errors: it is less wrong e.g. to assign a oocyte in state 1 to state 2 than to state 5, because the oocyte developments are more similar between states 2 and 1 than between states 5 and 1. Concerning ordinal problems, a common (but not totally correct) approach is to use nominal classifiers (obviating the ordinal information) or regressors (assuming that the distances between different categories are known and equal). Contrarily to these approaches, ordinal classifiers have been shown to achieve better performance (in terms of the class ordering) for multiple ordinal classification problems [13]. In the current paper, we test this hypothesis comparing the most outstanding ordinal classifiers (described

briefly in the following subsections) and nominal classifiers in the classification of developmental states of fish oocytes.

2.1 Threshold methods

Thresholds models assume that an underlying, unobservable real-valued outcome (the latent variable) exists for ordered crisp classes. These methodologies estimate: 1) a function $f(\mathbf{x})$ to predict the nature of the latent variable, i.e., a projection that maintains the classes ordered according to their rank; and 2) a vector of thresholds $\mathbf{b} = (b_1, b_2, \ldots, b_{K-1}) \in \mathbb{R}^{K-1}$ (where K is the number of classes) to represent the intervals in the range of $f(\mathbf{x})$, where $b_1 \leq b_2 \leq \ldots \leq b_{K-1}$. In our problem, a threshold model would try to uncover the latent variable related to the actual level of development of an oocyte, and the thresholds would divide this latent variable into the states of development considered. The first method in this category is the **Proportional Odds Model** (POM) [22], a reformulation of Logistic Regression for ordinal classification, wich links the cumulative probabilities to a linear predictor f and imposes a stochastic ordering of the input space. The ordinal version of Discriminant Learning, called **Kernel Discriminant Learning Ordinal Regression** (KDLOR) [29], constraints the classes to be ordered according to their ranking in the projection to optimize. Finally, the **Support Vector Ordinal Regression with IMplicit constraints** (SVORIM) is a reformulation of the Support Vector paradigm [6] which seeks for $K-1$ parallel separating hyperplanes to divide the data.

2.2 Decomposition methods

These techniques rely on the idea of decomposing the original ordinal problem into sets of simpler binary classification tasks [9, 30], which can be solved either by a single model or by a set of models. The subproblems are defined by a very natural methodology, considering whether a pattern $\mathbf{x}$ belongs to a class greater than a fixed k and combining the binary predictions in a unique ordinal class [20]. This idea has demonstrated very powerful for ordinal classification, in the same way as one-vs-one and one-vs-all approaches for nominal multiclass classification. The first decomposition method [9] computes $K-1$ binary classification models and relabels the dataset considering whether a pattern belongs to a class greater than a fixed k (which ranges from 1 to $K-1$). The posteriori output probabilities of each model are then fused to provide a unique ordinal prediction. Originally, the C4.5 decision tree classifier was used as the base binary methodology, but it has been recently demonstrated [30] that the SVM paradigm also leads to good performance for this purpose. Furthermore, it has been shown that the use of different weights per pattern (derived from the distances to the class k) helps to improve the performance. The combination of decomposed labels, weights per pattern and SVM base

methodology will be referred in the experimental section as **SVM with Ordinal Decompositions** (SVMOD). A reformulation of the Extreme Learning Machine, called **Extreme Learning Machine for Ordinal Regression** (ELMOR) [8], uses the one-of-K coding matrix for the outputs (commonly used with Artificial Neural Networks) and considers whether a pattern belongs to a class greater than a fixed k. Finally, the **Ensemble learning for ordinal regression with Product combiner and SVM** (EPSVM) combines binary and ternary classification tasks, trying to distinguish each class from the previous and subsequent ones and making use of a probability fusion function [25].

2.3 Reduction methods

These methods can also be seen as decomposition techniques, although with slight differences. The **REDuction SVM** (REDSVM) [20] transforms the training data $(\mathbf{x}_i, y_i)$ to extended data $(\mathbf{x}_i^k, y_i^k), 1 \leq k \leq K-1$, in such a way that $\mathbf{x}_i^k = (\mathbf{x}_i, k),\ y_i^k = 2[\![k < y_i]\!] - 1$, being $[\![\cdot]\!]$ a Boolean test which is 1 if the inner condition is true, and 0 otherwise, and using specific misclassification weights: $w_{y_i,k} = |c_{y_i,k} - c_{y_i,k+1}|$, where $\mathbf{C}$ is a cost matrix, with $c_{y_i,k-1} \geq c_{y_i,k}$ if $k \leq y_i$ and $c_{y_i,k} \leq c_{y_i,k+1}$ if $k \geq y_i$. Then, a binary classifier f is used with the extended data generating probabilistic values which are used to give an output prediction. The data replication method in [3], that will be referred as **Ordinal Neural Network** (ONN), represents a similar framework, except that it is based on a Multi-Layer Perceptron (MLP) neural network instead of SVM, being also less flexible because it assumes the absolute cost for the $\mathbf{C}$ matrix.

2.4 Ensemble based techniques

Opposed to the previous methods, some efforts have been made to derive a Boosting algorithm for ordinal regression by using thresholded ensemble models, with robustness for approximating complex labelling spaces [21]. This model is composed of confidence functions, and their weighted linear combination is used as the projection for the data. We tested two different approaches, **Ordinal Regression Boosting** (ORBoost) and **Ordinal Regression Boosting using Perceptrons** (ORBoostP), which use MLP neural networks and single perceptrons as base learners respectively.

3 Experimental work

In the following paragraphs we describe the data acquisition, validation methodology, methods tested and quality measures, discussing the results achieved.

3.1 Data acquisition

Subsamples of fixed ovaries were embedded in paraffin, sectioned at 3.5 μm and stained with *Haematoxylin-Eosin* standard protocol. We used *Leica*[2] hardware and software: a *DRE* research microscope to digitalize the histological sections, connected to a *DFC320* digital camera with *IM50* software, and the *Application Suite v.4.1* software to create mosaic images. The exposure time and color balance were set automatically. The spatial resolution at which the images were captured was 1.095 μm per pixel for species *Merluccius merluccius* (MC) and *Trisopterus luscus* (TL), and 3.943 μm per pixel for *Reinhardtius hippoglossoides* (RH). The outline of cells was manually drawn and classified by expert technicians using the Govocitos software. Color texture analysis relates the chromatic and textural information of images, providing good results in the classification of three states of development (CA, HYD and VIT/AT) for species MC and TL using nominal classifiers [12]. Govocitos uses a 25-length color-texture feature vector with 10 grey level texture features and 15 chromatic features. Grey level texture descriptors model the spatial relationship of a pixel and its neighbors, providing information of the image structure such as smoothness and regularity, among others. Specifically, we used the Local Binary Patterns [24], taking the uniform patterns with radius $R = 1$ and 8 neighbors. The chromatic features provide information about the distribution of the levels on each RGB channel, including the mean, variance, third an fourth statistical moments and entropy. The input patterns are preprocessed to have zero-mean and standard deviation one before being fed to the classifier (the mean and deviation values are calculated using only the training set).

3.2 Validation methodology

The data include patterns from species: 1) *Merluccius merluccius*: 1022 patterns with 3 states of development (classes): CA (25.3% of the total patterns), HYD (6.0%) and VIT/AT (68.7%). 2) *Trisopterus luscus*: 912 patterns with the same 3 states: CA (57.6%), HYD (1.5%) and VIT/AT (40.9%). Both species share the same experimental methodology: the data are divided in equal-sized training and validation sets, and for each classifier we select the values of the tunable parameters with the lowest Mean Absolute Error (MAE, see subsection 3.4) on the validation set. The test uses 5-fold cross validation. 3) *Reinhardtius hippoglossoides*: a set of 16 images (one image per individual, see Fig. 1) with 7915 cells and 6 states: PG (37.6%), CA (18.8%), VIT1 (20.8%), VIT2 (11.6%), VIT3 (8.0%) and VIT4 (3.2%). We performed two different experiments: a) Leave-One-Image-Out (LOIO): in each trial, a different image is excluded from training and used for test (this is the usual approach in image classification). The parameter tuning minimizes the MAE over 16 validation sets (50 patterns of each state), after training with 16 training sets (100 patterns of each state). The test results are achieved with the tuned classifier

[2] `http://www.leica.com`

averaging over the 16 test sets. b) Mixed images (MIX): we randomly selected 10 trios of training, validation and test sets, with 100, 50 and 50 patterns of each state respectively, belonging to all the images. The training and validation sets are used for parameter tuning as in LOIO, and the test performance is averaged over the 10 test sets. The whole data set is publically available[3].

3.3 Tested methods

Eleven ordinal approaches, described in the section 2, are tested: the linear method POM; different methods based on SVM: SVORIM, SVMOD, REDSVM and EPSVM; one method based on Discriminant Analysis: KDLOR; two methods based on Artificial Neural Networks concepts: ELMOR and ONN; and two ensemble models: ORBoost and ORBoostP. Additionally, we also compare to the well-known technique Support Vector Regression (SVR) in order to analyze whether a pure regression perspective could be suitable. All the SVM-based ordinal or nominal methods use LibSVM [5], tuning the regularization parameter C and the inverse γ of the kernel spread with values in $\{2^i\}_{-5}^{14}$ and $\{2^i\}_{-16}^{8}$ respectively. We use the sigmoidal activation function for ELMOR, ONN, ORBoost and ORBoostP, tuning the number of nodes in the hidden layer with values $\{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. For ONN and ORBoost, the number of hidden neurons was adjusted in the range $\{5, 10, 15, 20, 30, 40\}$. As proposed by [21], the number of ensemble members in ORBoost was 25 and 2000 for ORBoostP. The range of ϵ for ϵ-SVR was $\{10^i\}_0^3$. These ordinal methods are compared to the following thirteen nominal classifiers:

1. **ABR**: Adaboost.M1 ensemble of classification trees [10] implemented in the R language[4].
2. **ABW**. AdaBoostM1 ensemble of decision stump classifiers (one-node decision trees) implemented in Weka [14]. The percentage of weight mass for base training is tuned with values 25%, 50%, 75% and 100%.
3. **AvNN**: ensemble of five MLP neural networks trained with different random weight initializations, tuning the number of hidden neurons and the learning decay with values {1,3,5} and $\{0, 0.1, 10^{-4}\}$ respectively.
4. **BAG**: Bagging ensemble of decision trees [1].
5. **ELM**: Extreme Learning Machine [15], selecting the best activation function among sine, sign, hardlimit, triangular, radial basis and sigmoid functions, and tuning the number of hidden neurons with 20 values between 3 and 200. The inputs are scaled between -1 and +1, as recommended in the software documentation.
6. **GELM**: Gaussian kernel Extreme Learning Machine. The parameters C and γ are tuned similarly to LibSVM (see above).

[3] `https://wiki.citius.usc.es/datasets/fish_ovary`

[4] `http://www.r-project.org`

7. **GSVM**: Support Vector Machine (SVM) with Gaussian kernel, implemented as the remaining SVM-based methods (see above).
8. **LBR**. LogitBoost ensemble of decision stumps [11] with 200 Boosting iterations.
9. **LBW**. LogitBoost ensemble of decision stumps implemented in Weka [14], with the 100% of weight mass to train, five runs for internal cross-validation, shrinkage parameter $H = 1$ and 10 iterations.
10. **LDA**: The classical Linear Discriminant Analysis [27].
11. **MLP**: Multi-Layer Perceptron neural network, tuning the number of hidden neurons with values {1,3,5,7,9,11,13,15,17,19}.
12. **MLRM**: Multinomial Logistic Regression Model [19] with unlimited iterations and log-likelihood ridge 10^{-8}.
13. **RF**: Random Forest [2] ensemble of 500 trees, tuning mtry with values {2,4,7,9,12,14,17,19,22,25}.

Different software tools have been used to do so, such as the R language and the caret package[5], the ELM framework implemented in Matlab[6] and the Weka software [14].

3.4 Evaluation metrics

The first measure used to evaluate the previous classifiers is the well-known **Classifier Accuracy** (*Acc*, in %), the percentage of agreements between the desired and real classes without considering the class ordering. The **Cohen Kappa** (κ, in %), is based on *Acc* but discarding the probability of success by chance [4]. We also used other metrics specially designed for ordinal classification [7,13]. The **Mean Absolute Error** (MAE) is defined by $MAE = \frac{1}{N}\sum_{i=1}^{N}|r(y_i^*) - r(y_i)|$, being y_i^* and y_i the predicted and the true class respectively for pattern i, and $r(y)$ the rank of y (its position in the ordinal scale), being N the number of patterns. The MAE value ranges from 0 to $K-1$ (maximum deviation in the number of ranks between two labels). The last two metrics measure the correlation between predicted targets and true targets: the **Kendall tau rank correlation coefficient** (τ) measures the association between predicted and true class [17] as $\tau = \left(\sum_{ij} c_{ij}^* c_{ij}\right)\left(\sum_{ij} c_{ij}^{*2}\sum_{ij} c_{ij}^2\right)^{-1/2}$, where $i, j \in \{1, \ldots, N\}$, $c_{ij} = +1$ if $y_i > y_j$ (in the ordinal scale), being $c_{ij} = 0$ when $y_i = y_j$, and $c_{ij} = -1$ when $y_i < y_j$ (the same for c_{ij}^* using y_i^* instead of y_i). The τ values range from -1 (maximum disagreement between prediction and true label), to 0 (no correlation between them) and to 1 (maximum agreement). Finally, the **Spearman rank correlation coefficient** (ρ) is the Pearson correlation coefficient between the ranked predicted and true class [28], taking values in $[-1, 1]$ with the same significate as τ.

[5] http://caret.r-forge.r-project.org
[6] http://www.extreme-learning-machines.org

Table 1 Classification results: accuracy and Cohen κ (both in %), MAE, Kendall τ and Spearman ρ for species *Merluccius merluccius* and *Trisopterus luscus* with 3 states (CA, HYD, VIT/AT). Ordinal (resp. nominal) classifiers are in the upper (resp. lower) half of the table. The best and second best results are in bold and italics respectively.

Classifier	*Merluccius merluccius*					*Trisopterus luscus*				
	Acc.	κ	MAE	τ	ρ	Acc.	κ	MAE	τ	ρ
POM	87.8	73.1	0.164	0.800	0.834	92.0	83.7	0.140	0.856	0.865
KDLOR	85.4	71.1	0.160	0.832	0.870	84.8	72.9	0.166	0.854	0.894
SVORIM	89.5	78.6	0.132	0.839	0.874	94.7	89.1	0.095	0.902	0.908
SVMOD	**94.1**	**86.9**	**0.110**	**0.856**	0.858	94.9	89.7	0.097	0.900	0.901
ELMOR	93.2	85.4	0.116	0.848	0.859	91.5	83.0	0.152	0.842	0.850
EPSVM	89.0	74.0	0.161	0.777	0.828	92.3	84.2	0.143	0.855	0.858
REDSVM	89.1	77.0	0.140	0.834	0.861	93.8	87.4	0.112	0.886	0.890
ONN	86.8	73.6	0.162	0.810	0.853	91.5	83.8	0.112	0.889	0.911
ORBoost	90.1	79.0	0.124	*0.854*	**0.882**	93.1	86.5	0.096	0.905	0.922
ORBoostP	90.2	78.9	0.125	0.851	*0.878*	93.3	86.7	0.098	0.902	**0.917**
SVR	84.8	69.7	0.186	0.792	0.818	84.2	71.4	0.190	0.832	0.862
Mean	89.1	77.0	**0.144**	**0.827**	**0.856**	91.5	83.5	**0.127**	**0.875**	**0.889**
ABR	93.1	85.1	0.130	0.832	0.835	94.5	88.8	0.109	0.887	0.888
ABW	81.4	59.1	0.309	0.671	0.688	79.8	56.6	0.388	0.610	0.615
AvNN	93.6	86.1	*0.115*	0.851	0.854	**95.6**	**91.1**	*0.088*	*0.909*	0.910
BAG	91.8	81.8	0.148	0.806	0.811	91.0	81.5	0.179	0.815	0.815
ELM	93.5	85.8	0.119	0.845	0.849	94.6	89.0	0.105	0.892	0.893
GELM	*93.8*	86.4	0.119	0.845	0.847	95.2	90.1	0.093	0.904	0.906
GSVM	92.8	84.2	0.134	0.823	0.826	*95.5*	*90.9*	**0.087**	**0.911**	*0.912*
LBR	91.4	82.3	0.164	0.790	0.795	92.6	87.1	0.148	0.830	0.835
LBW	91.1	80.7	0.162	0.789	0.794	90.3	80.3	0.188	0.805	0.807
LDA	92.5	83.8	0.128	0.829	0.835	93.1	85.9	0.134	0.861	0.863
MLP	*93.8*	*86.5*	0.129	0.834	0.837	**95.6**	**91.1**	0.088	*0.909*	0.910
MLRM	93.3	85.4	0.127	0.835	0.837	93.4	86.8	0.127	0.869	0.871
RF	93.1	84.9	0.128	0.831	0.835	93.5	86.7	0.133	0.862	0.862
Mean	**91.9**	**82.5**	0.147	0.814	0.819	**92.7**	**85.1**	0.144	0.851	0.853

3.5 Results and discussion

Tables 1 and 2 report the results in terms of classification accuracy (Acc, %), κ, MAE, Kendall τ and Spearman ρ rank correlation coefficients for species MC and TL (Table 1) and for specie RH with LOIO and MIX methodologies (Table 2). The classifiers are divided into ordinal (upper part) and nominal (lower part). The highest Acc, κ, τ and ρ, and the lowest MAE, are highlighted for each specie and experiment (the second best value is italized). The mean values for the whole set of ordinal and nominal methods are also included. From the application point of view, almost all of the results are very promising both in Acc and MAE: for species MC and TL we achieve Acc=94.1%, MAE=0.110 and Acc=95.6%, MAE=0.087 respectively (this MAE value means that each state is misclassified with neighbor states less than 10%). For specie RH the best results are slightly worse in terms of Acc (67.8% and 80.4% for LOIO and MIX experiments), κ and MAE, but very similar for the τ and ρ correlation coefficients, where indeed the results are outstanding. This result could indicate that the states are properly ordered from a purely ranking perspective,

Table 2 Classification results for the specie *Reinhardtius hippoglossoides* with 6 states (PG, CA, VIT1, VIT2, VIT3, VIT4) using the Leave-One-Image-Out (LOIO) and Mixed Images (MIX) methodologies.

Classifier	Leave-One-Image-Out					Mixed Images				
	Acc.	κ	MAE	τ	ρ	Acc.	κ	MAE	τ	ρ
POM	63.9	49.5	0.394	0.770	0.851	70.5	64.6	0.310	0.879	0.940
KDLOR	66.7	52.9	0.360	0.789	0.856	78.9	74.7	0.222	0.912	*0.957*
SVORIM	67.2	53.5	*0.354*	0.788	0.858	79.0	74.8	0.222	0.911	*0.957*
SVMOD	**67.8**	**54.6**	**0.352**	*0.796*	**0.861**	*80.1*	*76.2*	*0.215*	0.912	*0.957*
ELMOR	62.7	49.0	0.418	0.762	0.831	77.5	73.0	0.247	0.898	0.948
EPSVM	66.9	53.5	0.358	**0.798**	*0.860*	75.6	70.7	0.259	0.897	0.950
REDSVM	67.2	53.4	0.362	0.789	0.856	78.7	74.4	0.225	0.910	*0.957*
ONN	63.7	49.4	0.393	0.768	0.851	74.3	69.1	0.276	0.890	0.945
ORBoost	64.5	50.2	0.376	0.772	0.849	77.6	73.1	0.232	0.909	*0.957*
ORBoostP	64.4	50.2	0.378	0.772	0.848	77.5	73.0	0.233	0.909	0.956
SVR	66.1	52.3	0.363	0.785	0.855	79.1	74.9	0.218	**0.914**	**0.959**
Mean	**65.6**	**51.7**	**0.373**	**0.781**	**0.852**	**77.2**	**72.6**	**0.242**	**0.904**	**0.953**
ABR	51.8	34.4	0.597	0.748	0.799	77.8	73.3	0.255	0.891	0.941
ABW	30.8	10.5	0.895	0.572	0.619	32.8	19.4	0.771	0.710	0.805
AvNN	53.0	37.8	0.626	0.772	0.822	79.3	75.2	0.229	0.905	0.952
BAG	50.6	35.0	0.675	0.714	0.771	70.7	64.8	0.382	0.824	0.889
ELM	64.4	50.2	0.424	0.750	0.808	78.3	73.9	0.249	0.894	0.942
GELM	65.6	51.7	0.400	0.766	0.821	*80.1*	76.1	0.221	0.908	0.953
GSVM	*67.4*	*53.7*	0.363	0.790	0.647	**80.4**	**76.5**	**0.212**	*0.913*	0.785
LBR	45.7	33.1	0.801	0.600	0.663	71.6	70.9	0.433	0.801	0.865
LBW	56.3	40.6	0.592	0.679	0.737	72.3	66.8	0.338	0.851	0.912
LDA	53.5	39.0	0.578	0.785	0.839	77.5	72.9	0.248	0.897	0.948
MLP	52.3	36.0	0.599	0.768	0.824	79.0	74.9	0.234	0.903	0.950
MLRM	66.1	53.0	0.372	0.794	0.848	78.3	73.9	0.237	0.903	0.951
RF	52.6	36.8	0.639	0.748	0.802	78.2	73.8	0.254	0.891	0.940
Mean	54.6	39.4	0.582	0.730	0.769	73.6	68.6	0.313	0.869	0.910

but the predictions might be displaced one or two values (recall that the correlation coefficients τ and ρ consider for example whether a pattern belonging to class $\mathcal{C}_2$ is ranked higher than a pattern belonging to class $\mathcal{C}_1$ but it does not consider whether they actually are included in this two states in the prediction, i.e. the pattern belonging to $\mathcal{C}_2$ could be predicted to belong to $\mathcal{C}_3$ and the one belonging to $\mathcal{C}_1$ to $\mathcal{C}_2$). The average MIX accuracy and κ are 12 and 21 points, respectively, above their LOIO counterparts (see the *Mean* row), because in the MIX experiments the oocytes of each image may be selected for the training or test set. Therefore, implicit information about the sample acquisition and processing is included in the training and test sets, which justifies better results compared to LOIO experiments (more realistic from the application point of view), where any information about an individual fish in the testing set is not included in the training set.

From the classifiers perspective, Tables 1 and 2 identify some outstanding methods: SVMOD (ordinal, the best for species MC and RH-LOIO and very near to the best in RH-MIX) and GSVM (nominal, very near to the best for specie TL and the best for RH in MIX experiments). Furthermore, the measures MAE, τ and ρ show the real difference between ordinal and nominal

more clearly than Acc and κ. A general conclusion from this table is that, although any ordinal or nominal classifier has chance of obtaining the best result, in general, ordinal classifiers perform better in mean in terms of measures (MAE, τ, ρ) that consider the ordinal nature of the dataset, sometimes at the expense of lower Acc values. The *Mean* rows show that for species MC and TL (3 states), the average Acc and κ are better for nominal classifiers (which do not consider the state order), while MAE, τ and ρ are better for ordinal classifiers. However, in species RH (6 states) the five measures are better for the ordinal classifiers. This suggests that the superiority of ordinal with respect to nominal classifiers increases with the number of states, being this superiority not so clear (at least, for non-ordinal quality measures) with fewer states. In fact, with 3 states is not probable for a nominal classifier to assign a pattern to a non-neighbor state, because 4 of 6 possible errors (corresponding to non-diagonal elements in the 3-order square confusion matrix) respect the state ordering. However, in specie RH (6 states) the probability of non-ordinal errors is biggest (20 of 30 possible errors involve non-neighbor states), enhancing the difference between ordinal and nominal clasifiers.

Table 3 Ranking results averaged over all the experiments (ordinal classifiers are in bold), ordered by increasing MAE ranking, and p-values of the T-test comparing the best classifier (SVMOD) and the remaining ones (significant differences for $p < 0.05$ are in bold face).

Position	Classifier	Acc.	κ	**MAE**	τ	ρ	p-value
1	**SVMOD**	**2.3**	**2.3**	**2.3**	**2.6**	**3.3**	—
2	**SVORIM**	8.0	8.3	*4.8*	*4.9*	*3.5*	0.261
3	GSVM	*4.0*	*4.0*	5.9	6.0	11.8	0.133
4	**ORBoost**	13.3	13.3	5.9	5.0	3.8	0.083
5	GELM	4.3	4.8	6.1	7.5	8.4	0.287
6	AvNN	6.8	6.8	6.5	5.6	7.1	0.372
7	**ORBoostP**	13.3	13.3	6.5	5.4	4.8	0.064
8	**KDLOR**	14.5	14.8	6.9	5.9	3.6	0.120
9	MLRM	8.5	8.3	7.0	6.4	7.6	**0.004**
10	**SVR**	14.5	14.8	8.1	7.4	6.8	0.137
11	**REDSVM**	9.8	10.3	8.3	7.6	6.4	**0.041**
12	MLP	7.3	7.0	8.4	7.6	8.3	0.331
13	**EPSVM**	14.5	14.3	9.0	8.9	8.3	**0.038**
14	LDA	14.0	15.0	9.5	9.3	9.6	0.210
15	**ELMOR**	13.8	13.8	9.8	10.4	10.1	0.058
16	ELM	8.3	8.0	10.4	11.1	11.8	0.133
17	ABR	12.5	12.8	11.4	11.6	12.1	0.249
18	RF	12.3	13.0	11.8	12.3	12.6	0.235
19	**ONN**	18.0	17.8	12.1	12.1	9.0	**0.024**
20	**POM**	18.3	18.8	13.9	13.8	12.4	**0.018**
21	LBW	17.8	18.0	14.9	16.0	15.9	0.052
22	BAG	19.0	19.0	15.1	15.4	15.5	0.094
23	LBR	18.0	15.8	16.5	16.9	16.8	0.132
24	ABW	24.0	24.0	17.8	17.8	17.6	**0.022**

Table 3 reports the ranking results obtained for each method and metric (averaged over all experiments) ordered by increasing MAE: the SVMOD performs properly for all the metrics, obtaining promising mean ranking values,

as well as SVORIM, which also obtains competitive results in terms of MAE, τ and ρ, and opposed to the nominal method GSVM, that obtains good ranking results only for Acc and κ. Since GSVM uses the one-vs-one paradigm, which is not designed to specifically minimise the ordinal errors, it presents a good general performance, but in terms of ordinal metrics it is generally worse. The last column of Table 3 shows the p-value of the T-test comparing the mean MAE value obtained by SVMOD to each one of the other classifiers: SVMOD is significantly better than 6 classifiers (MLRM, REDSVM, EPSVM, ONN, POM and ABW, whose p-value is in bold), with high p-values (lower difference) for AvNN, MLP, GELM, SVORIM, ABR, RF and LDA.

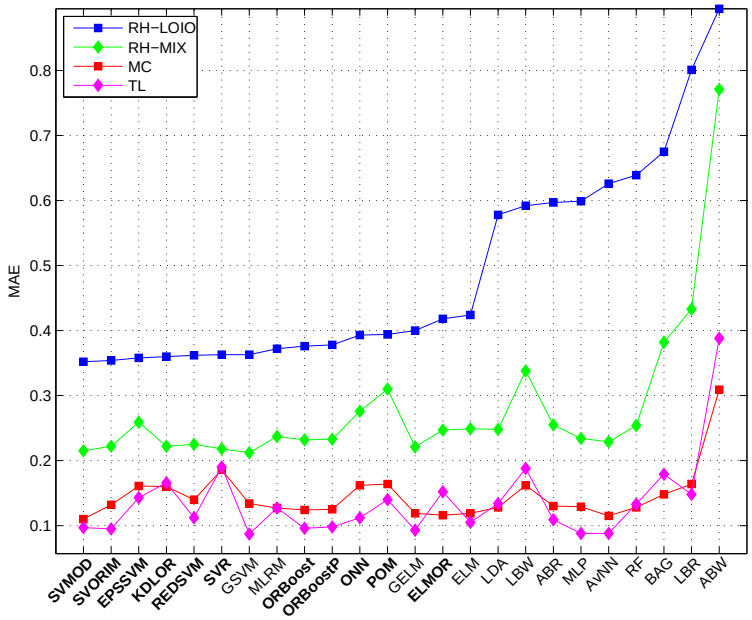


Fig. 2 MAE values achieved by each classifier (ordinal classifiers are in bold) for species RH (LOIO and MIX experiments), MC and TL, ordered by increasing the MAE for RH-LOIO.

Fig. 2 plots the mean MAE values of the classifiers for each experiment ordered by increasing MAE in RH-LOIO. Considering the LOIO plot (blue), the best accuracies are achieved by the ordinal classifiers (names in bold), which occupy most of the places in the left half of the horizontal axis. The nominal classifiers are in the right part of the axis, except GSVM and MLRM (7th and 8th positions respectively). There are clearly two groups: classifiers SVMOD to ELM (MAE about 0.35-0.40), and the remaining ones (LDA and following, MAE about 0.6 and higher). In the MIX experiments (green) many ordinal and nominal classifiers are below 0.25, and just a few are sub-optimals (EPSVM, ONN, POM, LBW, BAG and LBR). In specie MC (red) SVMOD is

the best, followed by GELM, ELMOR, ELM and AvNNet, while SVR, ONN, POM, LBW, BAG and LBR achieve bad results. In specie TL (magenta) the bests are GSVM, avNNet, MLP, GELM, SVORIM and SVMOD.

Table 4 Confusion matrices and sensitivities/positive predictivities for each state (in %) achieved by SVMOD (upper) and GSVM (lower) for specie RH and LOIO experiments.

SVMOD	PG	CA	VIT1	VIT2	VIT3	VIT4	Se(%)	PP(%)
PG	**22.29**	10.15	0.94	0.37	0.03	0.00	66.0	80.6
CA	5.00	**11.32**	2.29	0.09	0.01	0.00	60.5	45.3
VIT1	0.37	3.47	**14.23**	2.12	0.05	0.10	70.0	74.0
VIT2	0.00	0.04	1.68	**7.16**	0.92	0.15	71.9	68.3
VIT3	0.00	0.01	0.08	0.60	**6.00**	2.14	68.0	71.0
VIT4	0.00	0.00	0.02	0.14	1.44	**6.78**	80.9	74.0
GSVM	PG	CA	VIT1	VIT2	VIT3	VIT4	Se(%)	PP(%)
PG	**24.51**	7.66	1.17	0.29	0.14	0.00	72.5	80.3
CA	5.41	**10.36**	2.79	0.15	0.01	0.00	55.4	48.4
VIT1	0.58	3.35	**14.29**	2.02	0.03	0.08	70.2	71.6
VIT2	0.01	0.04	1.63	**7.00**	1.05	0.22	70.3	64.3
VIT3	0.00	0.00	0.05	1.23	**4.68**	2.88	53.0	62.5
VIT4	0.00	0.00	0.03	0.19	1.57	**6.59**	78.6	67.5

Table 4 reports the average confusion matrix, sensitivities (Se) and positive predictivities (PP) achieved by SVMOD and GSVM on specie RH with LOIO experiments (matrices for species MC and TL are not reported due to their low numbers of states). In both matrices the diagonal values are the highest in each row and column, and only the PP of state CA is below 50%, due to the overlap between states PG and CA (the largest non-diagonal values correspond to these two neighbor states). The only high non-diagonal values are adjacent to the diagonal, corresponding to patterns assigned to a state neighbor to the right one. Comparing SVMOD and GSVM, the latter achieves higher values outside the diagonal, excepting the (PG,CA) and (PG,VIT1) values, learning worse the ordinal information (remember from Table 2 that SVMOD wins GSVM with specie RH and LOIO experiments not only in Acc. and κ but also in MAE, κ and ρ). Besides, GSVM achieves lower Se and PP for all the states excepting PG, because it assigns more patterns CA to PG than SVMOD. In fact, the sensitivity of SVMOD is above 60% for all the states, while GSVM is below 55% for states CA and VIT3. Regarding PP, the SVMOD wins GSVM in all the states except CA, with high difference in states VIT1-VIT3.

4 Conclusions

This paper uses 11 ordinal and 13 nominal approaches to classify states of development of fish oocytes from histological microscopy images. Twenty-five features are extracted from every oocyte, including 10 grey level texture (Local Binary Patterns) and 15 statistical color features. Three fish species are considered: *Merluccius merluccius* and *Trisopterus luscus*, which present 3 states

of biological interest, and *Reinhardtius hippoglossoides*, with 6 states with and without leaving one image out. The experiments demonstrate that ordinal classifiers exhibit improved robustness and performance compared to nominal methods for all the species considered: SVMOD achieves accuracies about 94% and 95% for species MC and TL and 67%–80% for specie RH with and without leave one image out respectively. Several standard nominal techniques can also obtain promising results for some cases (GSVM for specie RH, without leave one image out, and AvNNet for specie TL). However, SVMOD has the best Friedman rank for all the five measures considered (Accuracy, Cohen κ, Mean Averaged Error, Kendall τ and Spearman ρ), and SVORIM is the second for the last three measures, which consider the ordinal nature of the classification problem (although GSVM is the second for Acc and κ). The difference between ordinal and nominal techniques has been shown to be higher when the number of states increases, being clearly reflected by ordinal quality measures (Kendall τ and Spearman ρ). The confusion matrix of SVMOD shows that the ordinal classifiers locate their errors in states near to the true ones, with sensitivities and positive predictions above 60% for almost all the states. On the whole, it can be said that ordinal regression techniques should be preferred to regression and multinomial classification methods when dealing with datasets that present an ordinal nature. This also motivates the improvement of the current techniques in the ordinal classification literature, which, given the novelty of the topic, are still in constant development.

References

1. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
2. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
3. Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: The data replication method. J. Machine Learning Research **8**, 1393–1429 (2007)
4. Carletta, J.: Assessing agreement on classification tasks: The kappa statistic. Computational Linguistics **22**(2), 249–254 (1996)
5. Chang, C., Lin, C.: LibSVM: a library for Support Vector Machines (2008). URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
6. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation **19**, 792–815 (2007)
7. Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.: Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing **135**(0), 21 – 31 (2014)
8. Deng, W.Y., Zheng, Q.H., Lian, S., Chen, L., Wang, X.: Ordinal extreme learning machine. Neurocomputing **74**(1-3), 447–456 (2010)
9. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proc. 12th Eur. Conf. on Machine Learning, pp. 145–156 (2001)
10. Freund, Y., Schapire, R.: Experiments with a new Boosting algorithm. In: Int. Conf. on Machine Learning, pp. 148–156. Morgan Kaufmann (1996)
11. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Annals of Statistics **28**, 2000 (1998)
12. González-Rufino, E., Carrión, P., Cernadas, E., Fernández-Delgado, M., Domínguez-Petit, R.: Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary. Pattern Recognition **46**, 2391–2407 (2013)

13. Gutiérrez, P., Pérez-Ortiz, M., Fernandez-Navarro, F., Sánchez-Monedero, J., Hervás-Martínez, C.: An experimental study of different ordinal regression methods and measures. In: 7th Int. Conf. on Hybrid Artificial Intelligence Systems (HAIS), *Lecture Notes in Computer Science*, vol. 7209, pp. 296–307 (2012)
14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The Weka Data Mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)
15. Huang, G., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. - Part B: Cybernetics **42**, 513–529 (2012)
16. Hunter, J.R., Macewicz, B.J., Lo, N., Kimbrell, C.A.: Fecundity, spawning and maturity of female Dover Sole, Microstomus pacificus, with an evaluation of assumptions and precision. Fisheries Bulletin **90**, 101–128 (1992)
17. Kendall, M.: A new measure of rank correlation. Biometrika **30**, 81–89 (1938)
18. Kennedy, J., Gundersen, A., Hoines, A., Kjesbu., O.: Greenland halibut (Reinhardtius hippoglossoides) spawn annually but successive cohorts of oocytes develop over 2 years, complicating correct assessment of maturity. Canadian Journal of Fisheries and Aquatic Sciences **68**, 201–209 (2011)
19. Le Cessie, S., Van Houwelingen, J.: Ridge estimators in Logistic Regression. Applied Statistics **41**(1), 191–201 (1992)
20. Li, L., Lin, H.T.: Ordinal regression by extended binary classification. In: Advances in Neural Information Processing Systems 19, pp. 865–872 (2007)
21. Lin, H.T., Li, L.: Large-margin thresholded ensembles for ordinal regression: Theory and practice. In: J. Balcázar, P. Long, F. Stephan (eds.) Algorithmic Learning Theory, *Lecture Notes in Computer Science*, vol. 4264, pp. 319–333. Springer Berlin Heidelberg (2006)
22. McCullagh, P., Nelder, J.A.: Generalized Linear Models, 2nd edn. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC (1989)
23. Morgan, M.J., Bowering, W.R.: Temporal and geographic variation in maturity at length and age of Greenland halibut (Reinhardtius hippoglossoides) from the Canadian North-West Atlantic with implications for fisheries management. ICES Journal of Marine Science **54**, 875885 (1997)
24. Ojala, T., Piatikäinen, M., Mäenpää, T.: Multiresolution grey-scale and rotation invariant texture classification with local binary pattern. IEEE Trans. Pattern Anal. Mach. Intell. **24 (7)**, 971–987 (2002)
25. Pérez-Ortiz, M., Gutiérrez, P.A., Hervás-Martínez, C.: Projection based ensemble learning for ordinal regression. IEEE Trans. on Cybernetics **44**(5), 681–694 (2014)
26. Rideout, R.M., Maddock, D.M., Burton, M.P.M.: Oogenesis and the spawning pattern in Greenland halibut from the North-west Atlantic. Journal of Fish Biology **54**, 196207 (1999)
27. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge Univ. Press (1996)
28. Spearman, C.: The proof and measurement of association between two things. Amer. J. Psychol **15**, 72–101 (1904)
29. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. IEEE Transactions on Knowledge and Data Engineering **22**, 906–910 (2010)
30. Waegeman, W., Boullart, L.: An ensemble of Weighted Support Vector Machines for Ordinal Regression. International Journal of Computer Systems Science and Engineering **3**(1), 1–7 (2009)

2.3. A system learning user preferences for multiobjective optimization of facility layouts

The efficiency of industrial production is widely influenced by the design of plant layouts. In fact, it is estimated that between 20 % and 50 % of production costs are due to materials handling, and that these costs can be reduced at least by 10 % and 30 % through efficient design. Most authors have solved this problem using quantitative criteria. Unfortunately, the approaches may not adequately consider all of the essential qualitative information that affects a human expert involved in design (e.g. engineers, stakeholders, regulators, etc).

In this paper, we try to solve this deficiency by making use of ordinal classification to construct a surrogate model which tries to imitate the decision maker. To do so, we construct a dataset composed of different facility layouts which are generated by an interactive genetic algorithm. Each layout is evaluated by an expert according to a Likert scale depending on his/her preferences. Several ordinal algorithms are tested, in combination with different cost matrices. The best model is then used as one of the fitness functions for a multiobjective evolutionary algorithm. The other fitness is computed considering an objective factor of each facility layout, in this case, the material flow cost. In this sense, the algorithm exploits the search space in order to obtain a satisfactory set of plant layouts. The proposal is applied on a design problem case where the classification algorithm demonstrated that it could fairly learn the user preferences, as the model obtained worked well guiding the search and finding good solutions. Moreover, the proposed system allows considering subjective designer preferences without reducing search extension. In many problems, like the one used in experimentation, considering these user preferences does not necessarily mean getting worse layout in terms of the other objective measures. Therefore, using the system proposed can have interesting results in terms of improving the quality of the facility layouts. The use of ordinal regression in this case allowed the integration of order information among labels in the model, and giving more importance to some kind of misclassification errors.

A system learning user preferences for multiobjective optimization of facility layouts

M. Pérez-Ortiz *, A. Arauzo-Azofra, C. Hervás-Martínez, L. García-Hernández, and L. Salas-Morera

University of Córdoba, Spain

Abstract. A multiobjective optimization system based on both subjective and objective information for assisting facility layout design is proposed on this contribution. A data set is constructed based on the expert evaluation of some facility layouts generated by an interactive genetic algorithm. This dataset is used for training a classification algorithm which produces a model of user subjective preferences over the layout designs. The evaluation model obtained is integrated into a multi-objective optimization algorithm as an objective together with reducing material flow cost. In this way, the algorithm exploits the search space in order to obtain a satisfactory set of plant layouts. The proposal is applied on a design problem case where the classification algorithm demonstrated that it could fairly learn the user preferences, as the model obtained worked well guiding the search and finding good solutions, which are better in term of user evaluation with almost the same material flow cost.

1 Introduction

Facility Layout Design (FLD) determines the placement of facilities in a manufacturing plant with the aim of determining the most effective arrangement in accordance with some criteria or objectives, under certain constraints. In this respect, Kouvelis et al. (1992) [11] provided that FLD is known to be very important for production efficiency because it directly affects manufacturing costs, lead times, work in process and productivity. According to Tompkins et al. (2010) [20], well laid out facilities contribute to the overall efficiency of operations and can reduce between 20% and 50% of the total operating costs. There are many kinds of layout problems. This contribution focus on the Unequal Area Facility Layout Problem (UA-FLP) as formulated by Armour and Buffa (1963) [3]. In short, UA-FLP considers a rectangular plant layout that is made up of unequal rectangular facilities that have to be placed effectively in the plant layout.

Aiello et al. (2012) [2] stated that, generally speaking, the problem of designing a physical layout involves the minimization of the material handling cost as the main objective. But, there are other authors that consider additional quantitative performance, as for example, Aiello et al. (2006) [1], who have addressed

*Partially subsidized by the TIN2011-22794 project of the spanish MICYT, FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

this problem taking into account criteria that can be quantified (e.g., material handling cost, closeness or distance relationships, adjacency requirements and aspect ratio), which are used in an optimization approach. However, Babbar-Sebens and Minsker (2012) [4] established that these approaches may not adequately represent all of the relevant qualitative information that affect a human expert involved in design (e.g. engineers). In this way, qualitative features sometimes also have to be taken into consideration. Brintup et al. (2007) [5] stipulated that such qualitative features are complicated to include with a classical heuristic or meta-heuristic optimization. Besides, according to Garcia-Hernandez et al. (2011) [10] these qualitative features can be subjective, not known at the beginning and can be changed during the process. As a consequence, the participation of the designer is essential to include qualitative considerations in the design. Moreover, involving the designer in the process provides additional advantages which have been detailed in its work.

The Interactive Genetic Algorithm (IGA) developed for FLD [10] consider user evaluation and handle subjective features. This algorithm uses a clustering mechanism to reduce the number of evaluations required from the user. However, running the IGA can be a tedious task for a designer, as many evaluations are still required. Fatigue is the main reason for an early stop of IGAs [14], thus reducing the possibilities of the system to find better designs. Moreover, user evaluation is some orders of magnitude slower than computed evaluation, leading necessarily to a much smaller search capacity. Learning user design preferences over a concrete layout problem would allow to simulate user responses. In this way, fatigue could be avoided and search could be performed much faster, which is specially useful in the context of the large search space of facility layouts. The goal of this contribution is to design a system that is able to learn these user layout preferences and perform a search considering both, the user preferences and other objective criteria.

For a layout design, the user evaluation considered in the IGA is of an absolute type deciding among five possible values for each design. From the user point of view, absolute evaluation is considered more practical than relative comparisons between layouts. Besides, absolute evaluation has shown better learning results when learning synthetic models of user evaluation [21].

Likert scales were firstly proposed in 1932 [13] as a way to produce attitude measures which could be interpreted as measurements on a proper metric scale. This technique is usually defined as a psychometric response scale, which is mainly used in questionnaires for obtaining the preference of different users or degree of agreement within a set of statements. In this paper, the most commonly likert scale is used in order to evaluate a set of facility layouts which have been synthetically created using an evolutionary algorithm. This rating technique can be seen as a 5-point (or granularity) scale ranging from "Strongly Disagree" on one and end to "Strongly Agree". Thus, the classes involved in the problem are: {*Strongly disagree, Disagree, Neither agree or disagree, Agree, Strongly agree*} where each class answers the question: *Could this plant layout be considered as a good solution for the Unequal Area Facility Layout Problem?*. Often, likert scale

items are treated numerically in such a way that it is assumed that the distance between all points on the scale are equal, however, this assumption might be wrong as we are forgetting the underlying latent variable in the scale. Because of that, optimal scaling is a relevant issue to both ordinal predictor and outcome variables. Likert scales can also be addressed from an ordinal regression point of view, where there is a certain order among class labels. Ordinal regression (or classification) is a relatively new learning problem which is rapidly growing and enjoying a lot of attention from pattern recognition and machine learning communities [9,18]. In this case, the classification problem is quite different to the standard one, as the variable to predict is not numerical or nominal, but ordinal, so categories have a natural order (in the same way that the categories we aim to predict in this paper). The major problem with this kind of classification is that there is not a precise notion of the distance between categories and there are some misclassification errors which should be more penalized.

This paper is organized as follows: a brief analysis of the system constructed is given in Section 2. All the experimental design features and the way the system learn the FLD preferences from the expert are presented in Section 3. Finally, we present some of the results obtained in Section 4 and Section 5 summarizes the conclusions and future work.

2 Structure of the proposed system

From a global perspective, the proposed system requires the designer to describe the problem and to evaluate several FLDs. At the end, the system must return a moderate number of designs according to the preferences found in the evaluation process and the optimization of the objectives factors. Thus, the main purpose would be combine both subjective and objective information in order to come to proper and fair decisions when evaluating these facility structures. To do so, a three-stage system has been developed as shown in Fig. 1:

1. Firstly, the IGA evolves towards FLDs that are preferred by the user. On each iteration, the user evaluates nine layouts which are stored for learning step. Although more layouts are generated and evaluated through clustering to better guide evolution, only user evaluated layouts are considered. Elicited evaluations from clustering may have been good for GA evolution but we have found that they may confuse the learning process.
2. After that, a machine learning classification algorithm will learn from the expert evaluations (using the dataset). In this stage, several nominal and ordinal algorithms are tested, in order to choose the one that achieved the best results. Once the more appropriated algorithm is selected, the model obtained is integrated in a multiobjective evolutionary algorithm, in such a way that this model will evaluate the facility layouts using the likert scale and this predicted target will be the first objective to maximize in the evolutionary learning process.
3. Finally, the second objective to maximize in the optimization will be an objective factor (in this case, the material flow between facilities). At this

point, the multiobjective algorithm is able to search and evolve through all possible solutions, and will end with a pareto front containing a set of optimal plant layouts.

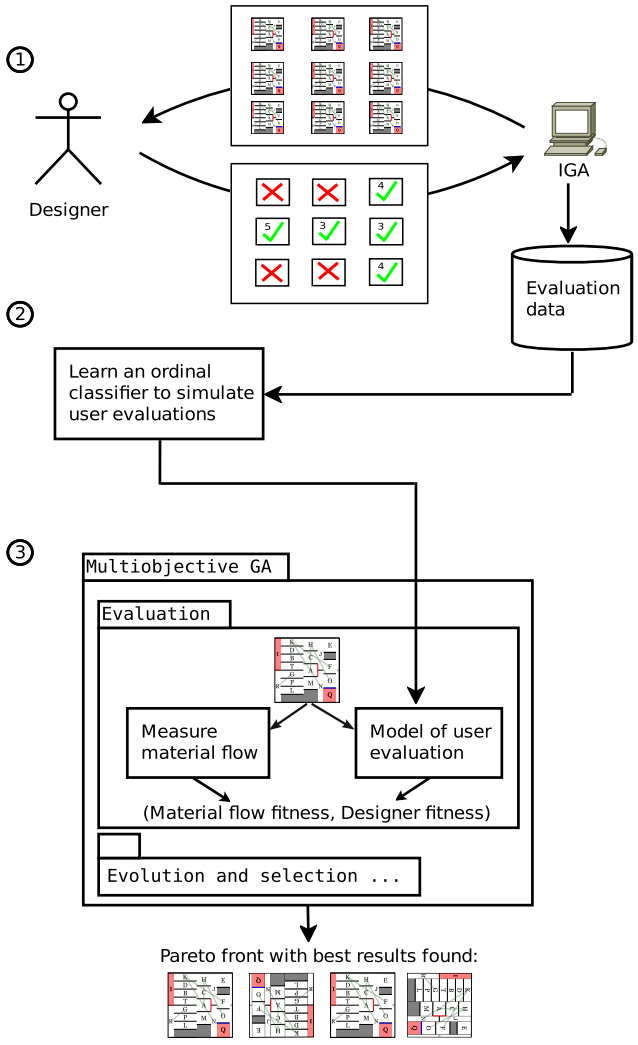


Fig. 1. Facility layout design system diagram

3 Learning facility layout preferences from the expert

Facility layouts from the IGA follow the Flexible Bay Structure (FBS) [19], where the facilities are placed in a series of rows of variable width. The data available to the learning algorithm is the number of bays and the geometrical coordinates of the rectangle assigned to each facility. Every of these layouts is tagged with an evaluation from a set of five ordered classes. Therefore, the user preferences will be learned with a supervised classification task. It must be noted that each problem case is different and usually will have different design preferences. For this reason, in principle, the knowledge and models created for one case are not applicable to others and a new model must be learned.

The facility layout problem case considered is composed of 20 facilities with different required areas that are arranged in a 61.7×61.7 meters plant. The IGA was run for 220 iterations using a population of 100 individuals with 0.5 probability of crossover and 0.05 probability of mutation. After removing some duplicated facility layouts, the IGA left us with a final database composed of 1969 patterns distributed in 5 classes and 86 attributes which contains information about the location and different characteristics of each facility distribution.

3.1 Selection of the algorithm

Several state-of-the-art methods have been tested for this problem in order to choose the one which performs better taking into account metrics which measures different kind of errors: CCR, which is the standard classification metric (also known as accuracy), MAE which measures ordinal classification and finally, MS which measures the worst classified class so this measure will help us to detect trivial classifications and will be really useful in unbalanced problems (as the one treated in this paper).

The Mean Absolute Error (MAE) is defined as the average deviation in absolute value of the predicted class from the true class. It is a commonly used for ordinal regression:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|,$$

where y_i is the true rank for the $i-th$ pattern, and $\hat{y}_i$ corresponds to the predicted rank for the same pattern. N is the number of test patterns.

The Minimum Sensitivity (MS) can be defined as the minimum value of the sensitivities for each class,

$$MS = \min\{S_i;\ i = 0, \ldots, K\},$$

where S_i is the sensitivity for the ith class. Sensitivity for class i corresponds to the correct classification rate for this specific class. In this way, MS reports the accuracy for the worst classified class.

In order to fairly compare the results obtained from different algorithms, a stratified 30-holdout and a nested 5-fold cross-validation have been performed. The algorithm tested for solving the given problem are the following ones:

– OCC (OrdinalClassClassifier): ensemble technique for ordinal regression [9] which applies C4.5 as base algorithm.
– KDLOR (Kernel Discriminant Learning for Ordinal Regression): method which combines discriminant analysis and kernel functions for ordinal regression [18].
– POM (Proportional Odd Model): one of the first models specifically designed for ordinal regression and arisen from a statistical background [15].
– EBC (Extended Binary Classification): ensemble method which performs multi-class classification with just a binary model [12]. Support Vector Machines are used as the base algorithm.
– SVC (Support Vector Classification): standard nominal classifier based on support vector machines which performs one-vs-one classification [7].
– C4.5: standard nominal decision tree [16] based on information entropy.

Table 1. CCR, MS, MAE obtained from the different methods

Algorithm	CCR	MAE	MS
OrdinalClassClassifier(C4.5)	*88.69 ± 1.48*	**0.169 ± 0.025**	*37.50 ± 9.51*
KDLOR	78.20 ± 2.19	0.231 ± 0.034	12.36 ± 1.87
POM	60.72 ± 1.24	0.468 ± 0.014	0.00 ± 0.00
EBC(SVM)	81.08 ± 2.76	0.249 ± 0.023	8.23 ± 2.50
SVC	80.15 ± 4.53	0.290 ± 0.047	13.81 ± 5.70
C4.5	**89.87 ± 2.31**	*0.175 ± 0.037*	**42.19 ± 7.35**

Taking into account these metrics, the best results in CCR and MS are achieved with the C4.5 algorithm [16], although it is not an ordinal method. But one can notice that best results in MAE are achieved using the OCC(C4.5). Nevertheless, as the differences are not significantly large and the algorithm selected will be integrated in a multiobjective algorithm, another important issue is the simplicity and one should take into account that the OCC is a ensemble model which makes use of probability functions for obtaining the final predicted targets. Because of that, the authors have considered the use of different ordinal cost matrices for training the C4.5 algorithm to see if the results could improve even more. This algorithm is also known as C4.5CS or cost-sensitive C4.5. C4.5CS [22] is a post-processor decision tree is used in conjunction with C4.5. This methodology implements cost-sensitive specialization by seeking to specialize leaves for high misclassification cost classes in favor of leaves for low misclassification cost classes. As said before, there are some misclassification errors which should be more penalized in the problem: confusing the "Strongly agree" with the "Strongly disagree" class should be considered by far a bigger mistake than confusing the "Neither agree or disagree" with the "Agree" class. Because of that, we have tested several approaches using cost matrices for solving this ordinal problem. The costs matrices used are shown in Table 2:

- Cost matrix #1: Usual cost matrix for the nominal classifiers, which assumes that all the misclassification errors are equal.
- Cost matrix #2: This matrix is the well-known standard ordinal one. It is widely used with nominal algorithms in order to weight the misclassification errors.
- Cost matrix #3: Quadratic ordinal cost matrix.
- Cost matrix #4: Cost matrix particularly proposed for the problem here addressed. Due to the fact that the best FLDs are a minority in the problem, authors have considered that it is critical not to miss any of them. Thus, errors when misclassifying an excellent plant layout are not the same as when misclassifying a bad one (because an expert will check over the final pareto front and will directly discard the non-proper ones).

Cost matrix #1					Cost matrix #2					Cost matrix #3					Cost matrix #4				
0	1	1	1	1	0	1	2	3	4	0	1	4	9	16	0	1	2	3	4
1	0	1	1	1	1	0	1	2	3	1	0	1	4	9	1	0	1	2	3
1	1	0	1	1	2	1	0	1	2	4	1	0	1	4	4	1	0	1	2
1	1	1	0	1	3	2	1	0	1	9	4	1	0	1	9	4	1	0	1
1	1	1	1	0	4	3	2	1	0	16	9	4	1	0	16	9	4	1	0

Table 2. Different cost matrices considered for the problem

The results obtained using these cost matrices with the same procedure as before (stratified 30-holdout) can be seen in Table 3 where one can notice that an important improvement of the results is produced by using these ordinal cost matrices. The cost matrix #4 is the one hybridized with the ordinal standard and the quadratic one, and it obtains the optimal results. Besides it is the one in concordance with the misclassifying errors to avoid in the problem, so it will be used for computing the final model to guide the evolutionary searching process.

Table 3. CCR, MS, MAE obtained with C4.5 and the different cost matrices

Cost matrix	CCR	MAE	MS
C4.5 (cost matrix #1)	88.69 ± 1.48	0.175 ± 0.037	42.19 ± 7.35
C4.5 (cost matrix #2)	89.82 ± 1.19	0.181 ± 0.014	51.56 ± 4.87
C4.5 (cost matrix #3)	*90.17* $\pm$ *1.24*	*0.173* $\pm$ *0.026*	*53.12* $\pm$ *6.56*
C4.5 (cost matrix #4)	$\mathbf{90.35 \pm 1.56}$	$\mathbf{0.163 \pm 0.021}$	$\mathbf{54.69 \pm 3.42}$

At this point, the procedure to train the final model have been established: use the C4.5 algorithm with a hybrid and ordinal cost matrix and the entire dataset (without any training and testing partitioning) to obtain a single final model.

4 Combining subjective and objective criteria

Once a model of the evaluation according to design preferences is learned. It is also desired that the facility layout optimize other objective features. Material flow is considered a very important measure of a good FLD. It is calculated as the product of the distance and the material movement estimated between facilities. In order to find a good facility layout considering both objectives, a multi-objective genetic algorithm (MOGA) is applied in the last stage of the system.

NSGA2 [8] algorithm has been used because it applies well known important theory on multi-objetive evolutionary algorithms [6] and achieves good results in facility layout problems [2]. The proposed system includes NSGA2 implemented using DEAP [17]. The encoding scheme uses a permutation of facilities and a binary vector with the points where bays are split. Crossover operators are PMX for facilities and two point crossover for split points, while mutation swaps the position of two facilities or toggles a bit in the split points vector.

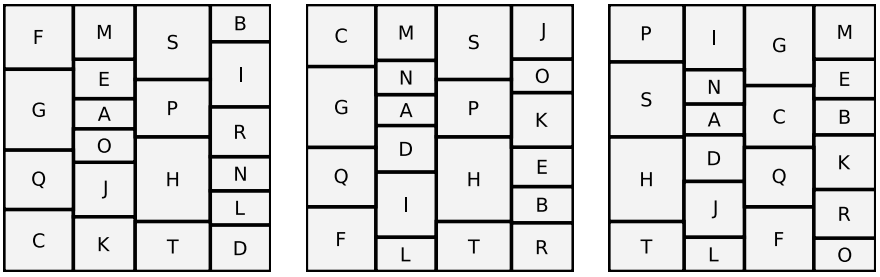
Apart from optimizing material flow and designer preferences, facilities of a given area must also have an usable shape that allows allocation of machines or other resources. This is usually controlled with aspect ratio constraints. Rather than discarding infeasible solutions according to aspect ratio constraints, a penalty function is used. In this way, some infeasible solutions are preserved to allow convergence to solutions that lie in the boundary between feasible and infeasible solutions [6]. We considered including this penalty as an additional objective. However, we have achieved better results using the adaptive penalization over material flow by Tate et. al. [19].

Finally, the system returns the pareto front with at most five solutions (one for each user evaluation class considered). The designer can now choose the facility layout with the optimum equilibrium between material flow and the other subjective preferences.

4.1 Case results

The described MOGA has been run 500 generations with a population of 200 individuals, using probabilities of 0.8 for crossover and 0.2 for mutation. Figure 2 shows the final results obtained by the system for this problem. There are no solutions with user evaluation values of 1 or 4 because all of them have been dominated by the ones shown. While material flow is pretty similar for all three layouts, the satisfaction of the user is much better in the third one. In this way, the proposed system improves the results of automatic facility layout because user preferences are included in the search without losing the fitting of the objective material flow measure.

In order to compare these results with those achieved by other algorithms not considering user preferences, a well known algorithm proposed by Aiello [1] has been run on the same problem with the same parameters. The best FLD found by this algorithm has a material flow of $189463u*m$ and it is necessary to increase the number of generations to reach FLDs with a similar material flow to



a) Material flow of $164766u * m$ and 2 as user evaluation

b) Material flow of $164782u * m$ and 3 as user evaluation

c) Material flow of $164799u * m$ and 5 as user evaluation

Fig. 2. Final pareto front with the FLDs found by the system

those obtained with the MOGA proposed. Intuitively, this may mean that user evaluation is helping to direct the search. This opens another research question requiring further experimentation with more data on diverse FLD problems.

5 Conclusions and future work

This paper proposes the construction of a system which combines user preferences and other objective factors. The proposed system allows considering subjective designer preferences without reducing search extension. In many problems, like the one used in experimentation, considering these user preferences does not necessarily mean getting worse layout in terms of the other objective measures. So, using the system proposed can have interesting results in terms of improving the quality of FLDs.

The use of ordinal regression in this case allowed us the integration of order information among labels in the model, and giving more importance to some kind of misclassification errors. Concerning future work, the system could be constructed by jointly taking into account the preferences of several users.

References

1. Aiello, G., Enea, M., Galante, G.: A multi-objective approach to facility layout problem by genetic search algorithm and electre method. Robotics and Computer-Integrated Manufacturing 22, 447–455 (2006)
2. Aiello, G., Scalia, G.L., Enea, M.: A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. Expert Systems with Applications (0) (2012)
3. Armour, G.C., Buffa, E.S.: A heuristic algorithm and simulation approach to relative location of facilities. Management Science 9, 294–309 (1963)
4. Babbar-Sebens, M., Minsker, B.S.: Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design. Applied Soft Computing 12(1), 182 – 195 (2012)

5. Brintup, A.M., Ramsden, J., Tiwari, A.: An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. Computers in Indust. 58, 279–291 (2007)
6. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.v.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, 2nd ed. edn. (Oct 2007)
7. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University, 1 edn. (2000)
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. In: PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature. pp. 849–858. Springer-Verlag, London, UK (2000)
9. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proc. of the 12th Eur. Conf. on Machine Learning. pp. 145–156 (2001)
10. Garcia-Hernandez, L., Salas-Morera, L., Arauzo-Azofra, A.: An interactive genetic algorithm for the unequal area facility layout problem. In: 6th Int. Conf. Soft Computing Models in Industrial and Environmental Applications SOCO 2011, vol. 87, pp. 253–262. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
11. Kouvelis, P., Kurawarwala, A.A., Gutierrez, G.J.: Algorithms for robust single and multiple period layout planning for manufacturing systems. European Journal of Operational Research 63(2), 287–303 (1992)
12. Li, L., Lin, H.T.: Ordinal Regression by Extended Binary Classification. In: Advances in Neural Information Processing Systems 19 (2007)
13. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology 22(140) (1932)
14. Llor, X., Sastry, K., Goldberg, D.E., Gupta, A., Lakshmi, L.: Combating user fatigue in iGAs: partial ordering, support vector machines, and synthetic fitness. In: Proceedings of the 2005 conference on Genetic and evolutionary computation. p. 13631370 (2005)
15. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Monographs on Statistics and Applied Probability, Chapman & Hall/CRC, 2nd edn. (1989)
16. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
17. Rainville, F.M.D., Fortin, F.A., Gardner, M.A., Parizeau, M., Gagn, C.: Distributed evolutionary algorithms in python (deap). http://deap.googlecode.com (2011)
18. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. IEEE Transactions on Knowledge and Data Engineering 22, 906–910 (2010)
19. Tate, D.M., Smith, A.E.: Unequal area facility layout using genetic search. IIE Transactions 27, 465–472 (1995)
20. Tompkins, J., White, J., Bozer, Y., Tanchoco, J.: Facilities Planning. Wiley, New York, 4rd ed. edn. (2010)
21. Wang, S., Wang, X., Takagi, H.: User fatigue reduction by an absolute rating data-trained predictor in IEC. In: IEEE Conf. on Evolutionary Computation, 2006. pp. 2195 –2200 (2006)
22. Webb, G.I.: Cost sensitive specialisation. In: Foo, N., Goebel, R. (eds.) Lecture Notes in Computer Science Vol. 1114. Topics in Artif. Intel.: Proc. of the Fourth Pacific Rim Intern. Conf. on Artificial Intelligence (PRICAI'96). pp. 23–34. Springer-Verlag (1996)

That is what learning is. You suddenly understand something you've understood all your life, but in a new way.

Doris Lessing

3

Labelling decomposition methods for ordinal regression

This chapter presents different research works concerning decomposition methods, including new proposals and a thorough set of experiments. The chapter also includes two applications of these methods to real-world problems.

Main publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás Martínez. Projection-Based Ensemble Learning for Ordinal Regression. *IEEE Transactions on Cybernetics*, 44(5):681–694, 2014, Impact Factor (2013): 3.781 (Q1).
- M. Pérez-Ortiz, M. de la Paz-Marín, P.A. Gutiérrez and César Hervás-Martínez. Classification of EU countries' progress towards sustainable development based on ordinal regression techniques. *Knowledge-Based Systems*, 66:178–189, 2014, Impact Factor (2013): 3.058 (Q1).
- M. Pérez-Ortiz, M. Cruz-Ramírez, M. D. Ayllón-Terán, N. Heaton, R. Ciria and C. Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. *Applied Soft Computing*, 14:88–98, 2014, Impact Factor (2013): 2.679 (Q1).

- M. Pérez-Ortiz, P.A. Gutiérrez y C. Hervás-Martínez. Log-gamma distribution optimisation via maximum likelihood for ordered probability estimates. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8480, pages 454–465, 2014.

Other publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, J. Briceño y M. de la Mata. An ensemble approach for ordinal threshold models applied to liver transplantation. In *International Joint Conference on Neural Networks* (IJCNN), pages 2795–2802, 2012.

The four main publications are now presented in the different subsections of this chapter.

3.1. Projection-based ensemble learning for ordinal regression

Ordinal information gives us the possibility of comparing the different labels. For a given rank k, a direct question could be the following, "is the label of pattern $\mathbf{x}$ greater than k?". This is exactly the main motivation for most decomposition methods in ordinal classification. These techniques are based on the simplification of the original task through the formulation of several order hypotheses. The following paper presents a slightly different approach in this line. It could be said that the proposal is a reformulation of the one-versus-all scheme to ordinal classification. Every single model is trained in order to distinguish between a given class (k) and all the remaining ones (but grouping these in those classes with a rank lower than k, and those with a rank higher than k).

Three different base methodologies are tested in this case for the construction of the separate models: kernel discriminant learning, support vector machines and logistic regression (all reformulated to deal with ordinal regression problems). All of these methods share one common property, the fact that they are threshold methods which compute a projection and project the data for its subsequent classification. From this projected data, posterior probabilities of class belonging can be extracted and these could be very useful for constructing an accurate ensemble. Different weighting strategies and combiners are also explored in this paper in order to analyse the best combination.

The results show that the method is seen to be competitive when compared with other state-of-the-art methodologies (both ordinal and nominal), by using six measures and a total of 15 ordinal datasets. The superiority of the proposal with respect to the one-versus-all standard paradigm has been confirmed when dealing with ordinal regression. Although multiclass imbalance problems pose important difficulties for machine learning

algorithms, this formulation considered seems to achieve not only good global performance, but also good error rates for all classes independently, as demonstrated with several evaluation metrics. Furthermore, an additional set of experiments is conducted to demonstrate the potential scalability and interpretability of the proposed method when using logistic regression as base methodology for the ensemble.

Projection-Based Ensemble Learning for Ordinal Regression

María Pérez-Ortiz, Pedro Antonio Gutiérrez, *Member, IEEE,* and César Hervás-Martínez, *Member, IEEE*

***Abstract*—The classification of patterns into naturally ordered labels is referred to as ordinal regression. This paper proposes an ensemble methodology specifically adapted to this type of problem, which is based on computing different classification tasks through the formulation of different order hypotheses. Every single model is trained in order to distinguish between one given class (k) and all the remaining ones, while grouping them in those classes with a rank lower than k, and those with a rank higher than k. Therefore, it can be considered as a reformulation of the well-known one-versus-all scheme. The base algorithm for the ensemble could be any threshold (or even probabilistic) method, such as the ones selected in this paper: kernel discriminant analysis, support vector machines and logistic regression (LR) (all reformulated to deal with ordinal regression problems). The method is seen to be competitive when compared with other state-of-the-art methodologies (both ordinal and nominal), by using six measures and a total of 15 ordinal datasets. Furthermore, an additional set of experiments is used to study the potential scalability and interpretability of the proposed method when using LR as base methodology for the ensemble.**

***Index Terms*—Discriminant analysis, ensemble, logistic regression, ordinal classification, ordinal decomposition, ordinal regression, support vector machines, threshold models.**

I. Introduction

ORDINAL REGRESSION can be defined as a relatively new learning paradigm whose aim is to learn a prediction rule for ordered categories. This problem, firstly arising in statistics [2], is spreading rapidly and receiving a lot of attention from the pattern recognition and machine learning communities [3], [4] because it presents a wide range of applications in areas where human evaluation plays an important role, for example: psychology, medicine, information retrieval, etc. The main difference compared to standard regression is in the target variable, which is composed of finite and discrete category labels, the distances between them being unknown. Concerning classification, the variable to predict is not numerical or nominal, but ordinal; thus these categories show an implicit and natural order. An explanatory example of order among categories could be the Likert scale, a well-known methodology used for questionnaires, where the categories correspond to the level of agreement or disagreement with a series of given statements. The scheme of a typical five-granularity Likert scale could be: {Strongly disagree, Disagree, Neither agree or disagree, Agree, Strongly Agree}, where the natural order among categories can be appreciated. The major problem within this kind of classification is that misclassification errors should not be treated equally: misclassifying the Strongly disagree class as Strongly agree should be more penalized than misclassifying it as Disagree. Therefore, several issues must be taken into account in order to exploit the presence of this order among categories. First, this implicit data structure should be learnt by the classifier in order to minimize ordinal classification errors and, second, several measures or metrics should be developed in order to do so, given that simply being accurate might not be enough for this kind of problem.

Several approaches to tackle ordinal regression have been proposed in the domain of machine learning over the years, since the first work dating back to 1980 [2]. The simplest idea is to transform these ordinal scales into numeric values and solve the problem as a standard regression one. Kramer *et al.* [5] investigated and proposed the use of a regression tree learner in this sense. However, as outlined before, there is an important problem within these approaches: the fact that, in general, there is no knowledge about the distances between different classes. On the other hand, other works focused on addressing the problem by simply performing multinomial classification tasks (totally forgetting the order information) or by considering cost-sensitive classification [6] based on trivially imposed cost matrices. Some researchers approach the problem by decomposing the original ordinal regression task into a set of binary classification tasks [3], [7], or by formulating the original problem as one of extended binary classification [8], [9]. However, the most popular approach is clearly the use of threshold models [4], [10]–[12]. These methods are based on the idea that, in order to model ordinal classification problems from a regression perspective, one can assume that some underlying real-valued outcomes exist (also known as latent variable), although they are unobservable. Consequently, these methodologies estimate:

1) a function $f(\mathbf{x})$ that tries to predict the nature of those underlying real-valued outcomes;
2) a set of bias terms or thresholds $\mathbf{b} = (b_1, b_2, \ldots, b_{K-1}) \in \mathbb{R}^{K-1}$ (where K is the number of

Manuscript received October 29, 2012; revised May 1, 2013; accepted May 30, 2013. Date of publication June 27, 2013; date of current version April 11, 2014. This work was supported in part by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds, and the P2011-TIC-7508 project of the "Junta de Andalucía," Spain. This paper was recommended by Associate Editor N. Chawla.

The authors are with the Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Córdoba 14071, Spain (e-mail: i82perom@uco.es; pagutierrez@uco.es; chervas@uco.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2266336

classes in the problem) to represent the intervals in the range of $f(\mathbf{x})$, where $b_1 \leq b_2 \leq \cdots \leq b_{K-1}$.

Nowadays, the ensemble paradigm is one of the most actively researched in pattern recognition and machine learning [13]. This methodology imitates human nature to seek several opinions before making a crucial decision [14] and was proposed as an alternative to the conventional standalone methods, which can be suboptimal. The main aspects addressed in ensemble literature are: development of methods for reducing the dependence between classifiers, i.e., maximizing diversity, and development of effective combination rules.

This paper contributes a novel and natural ensemble methodology to tackle ordinal information which could be used with any threshold model as base classifier. More specifically, in this paper kernel discriminant analysis (KDA) [4], [15] and support vector machines (SVMs) [16], [17] were used for a first set of experiments, since these can be considered accurate and successful methods when adapted to ordinal regression [18], [19]. Moreover, logistic regression (LR) [2], [20] was considered for a set of large-scale datasets. The main motivation is the development of an ordinal ensemble algorithm which could benefit from the order information of the data to improve the performance of other existing techniques. As many classifiers as the number of classes are trained, and each single model is computed to differentiate each class from the remaining ones taking ordinal ranks into account, i.e., separating each class from the previous and following classes. The ensemble methodology proposed is based on decomposing ordinal regression problems into simpler classification tasks, where the order information is explicitly included. For a K class ordinal regression problems, two binary classification problems and $K-2$ ordinal ones (each composed of three classes) are derived, in such a way that the main classification problem is simplified. This procedure can be appreciated in Fig. 1 for a 5 classes example. The main hypothesis is that the performance of any ordinal algorithm could be improved by simplifying classification tasks and formulating multiple order hypotheses which will be combined in a final decision function. The proposal can be seen as a reformulation of the one-versus-all idea to tackle ordinal regression. A set of experiments is presented in this paper, which tests and validates this methodology and other nominal and ordinal ones, taking into account 15 datasets with different characteristics. The results suggest that the proposal reaches a competitive performance level and is able to extract better quality classifiers from the order information in the class labels. Finally, a different set of experiments over two large-scale datasets is conducted to analyze the potential scalability and interpretability of the proposed ensemble.

This paper is a very significant extension of [1] with much additional material, including a comprehensive review of some ordinal regression methodologies, a more detailed description of the proposal with some changes, and a wider experimental section, where the results for different benchmark datasets and measures were analyzed. Besides, SVMs and logistic regression techniques formulated for ordinal regression were also considered in this paper, both for ensemble construction and for comparison.

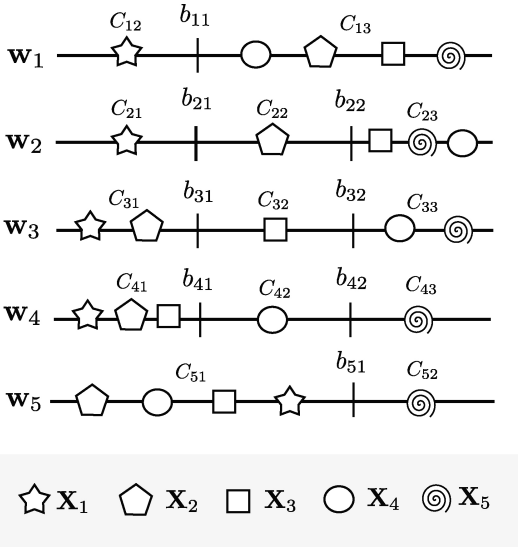


Fig. 1. Example showing different projections computed for the ensemble when $K = 5$. $\mathbf{X}_i$ are the patterns associated to class i. The model trained for separating class ith from the remaining ones is denoted by $\mathbf{w}_i$ and the corresponding thresholds associated by b_{i1} and b_{i2}. C_{ij} is used for denoting a synthetically constructed cluster of classes for decision maker ith.

Some advantages and decisions related to the proposal are now discussed. First, the choice of threshold models as base classifiers is justified because of their inherent advantage to lend themselves to probabilistic outputs, as these conditional probabilities of class membership are useful for constructing a more robust ensemble methodology. The proposal can be applied to any threshold model (indeed to any algorithm leading to probabilistic outputs), since the main idea is to compute one model to differentiate each class from the rest by taking ordinal ranks into account, and then extracting final output probabilities from the outcomes of each model. In addition, threshold methods depend to a great extent on the bias or threshold computation, which may be a complex handicap when dealing with kernel methods because of their tendency to over-fit. Instead of using crisp values, this paper considers probability estimations to relax and alleviate the misclassification error of multiple order hypotheses. On the other hand, selecting the number of classifiers has always been one of the most important and controversial issues in the ensemble paradigm (this value is usually assigned to an odd number in order to avoid draws), but in this case it is very intuitive, as the number of classifiers would be preassigned to the number of classes in the sample. Also, inducing diversity in the classifiers is a crucial ingredient for developing robust ensemble techniques. However, in this case diversity is implicit in the technique, as each computed model will be composed of different data labeling and pattern distributions. Finally, the proposal could also be justified by the low number of ordinal ensemble methods existing in the literature.

The paper is organized as follows. Section II shows a description of the methodologies used for the ensemble; Section III formally presents the proposal of this paper;

Section IV describes the characteristics of the datasets and the experimental study and analyzes the results obtained; and finally Section V outlines some conclusions and future work.

II. Previous Notions

In this section, the terminology and notation that will be used throughout the entire paper is established. The goal in classification is to assign an input vector $\mathbf{x}$ to one of K discrete classes $\mathcal{C}_k$, where $k \in \{1, \dots, K\}$. Thus, a formal framework for the ordinal regression problem could be introduced by considering an input space $X \in \mathbb{R}^d$, where d is the data dimensionality. To do so, an outcome space $Y = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ is defined, where the labels are ordered due to the data ranking structure ($\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$, where $\prec$ denotes this order information). Let N be the number of patterns in the sample and N_k the number of samples for the kth class. The objective in this kind of problem is to find a prediction function $f : X \to Y$ by using an i.i.d. sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in X \times Y$.

The ensemble approach here proposed is applied to three well-known techniques: KDA, SVM, and LR. Since they have been reformulated to deal with ordinal regression problems, a brief explanation of these methods is included in this section.

A. Kernel Discriminant Learning

This learning paradigm (KDA) is one of the pioneer and leading techniques in the machine learning area, since it dates back to 1936 and has been widely used as much for supervised dimensionality reduction as for classification [21]. KDA has also been adapted to ordinal classification [4] by imposing a constraint on the projection to be computed, so that it will preserve and take advantage of the ordinal information from different classes. The method is known as kernel discriminant learning for ordinal regression (KDO) [4].

B. Support Vector Machines

The SVM paradigm [16], [22] is considered to be the most common kernel learning method for statistical pattern recognition. This paper considers two of the most commonly used approaches for solving multiclass problems with SVMs: the one-versus-all formulation and the one-versus-one formulation.

Some works in the SVM literature have been focused on the reformulation of this successful paradigm to tackle ordinal regression problems [17], [23], [24]. All these approaches share one common objective which is the definition of $K-1$ discriminant hyperplanes represented by the vector $\mathbf{w}$ and the scalars bias $b_1 \leq \cdots \leq b_{K-1}$ in order to properly separate training data into ordered classes by modeling ranks as intervals on the real line.

The proposal of Herbrich [23] derived the well-known SVM methodology for ordinal regression by making use of an independent distribution model and inducing an ordering in the space X that incurs the smallest number of inversions on pairs $(\mathbf{x}_i, \mathbf{x}_j)$ of objects, the probability of that incurred inversion being given by a risk function for each pair of ranks. The main disadvantage of this algorithm is that the problem is formulated as a quadratic function directly depending on the training number of patterns.

On the other hand, the work of Shashua and Levin [24] introduced two different approaches: the former tries to maximize the margin between the closest neighboring classes by applying the "fixed margin" policy and the latter allows for different margins where the sum of margins is maximized. The principal disadvantage of their proposal is that ordinal inequalities on the thresholds, $b_1 \leq b_2 \leq \cdots \leq b_{K-1}$, are not included in the formulation and this omission may result in disordered thresholds at the solution.

A third proposal of SVMs for ordinal regression is presented in the work of Chu and Keerthi [17]. This study also shows two different implementations for the idea. Both approaches guarantee that the thresholds are properly ordered at the optimal solution. The first one only takes into account adjacent ranks for the determination of the thresholds, whereas in the second one, the whole training sample considering all ranks is used for the determination of each threshold, and samples in all the categories are allowed to contribute errors for each hyperplane. This second approach is called support vector ordinal regression with implicit constraints (SVOI).

From another point of view, ordinal regression can be transformed into several binary classification problems; one binary classifier can be derived for each problem, and the output of all classifiers can be combined to obtain a final decision. The strategy is based on simply checking if the rank of a pattern is greater than a given rank k, $1 \leq k \leq K-1$, which is indeed a binary classification question which is answered by each classifier. This approach is closely related to that proposed in this paper and was first presented in the work of Frank and Hall [3] with C4.5 classification trees as base classifiers. However, SVMs have performed very competitively for binary problems, and a similar proposal was then considered for SVMs in the work of Waegeman and Boullart [7], but introducing specific weights into the different patterns. These weights try to reflect the fact that not all patterns in the greater than k class (for the binary classifier k) are equally far from k in the ordinal scale, and they should be treated differently when constructing the classifier (even though they belong to the same class). Both methods will be considered in the experimental section.

C. Logistic Regression

In machine learning, LR [20] is a well-known methodology based on a regression analysis for classification problems. This method has been reformulated to deal with ordinal problems giving rise to the proportional odds model (POM) [2]. This model was the first threshold method applied to ordinal regression problems and it is based on a linear projection jointly trained with a set of thresholds by using a similar technique to that considered for nominal LR. Let h denote an arbitrary monotonic link function. The model

$$h\left(P(y \leq \mathcal{C}_j|\mathbf{x})\right) = \mathbf{w}^\top \mathbf{x} - b_j, \quad j = 1, \dots, K-1 \qquad (1)$$

links the cumulative probabilities to a linear predictor and imposes an stochastic ordering of the space $\mathcal{X}$, where b_j is the threshold separating $\mathcal{C}_j$ and $\mathcal{C}_{j+1}$ and $\mathbf{w}$ is a linear projection.

III. Ensemble Learning for Ordinal Regression

In the previous section, three well-known classification methods have been presented: KDA, SVM, and LR. These methods share one common and general objective that defines the optimization function: the maximization of the distance between different classes. Therefore, they depend greatly on the number of classes in the sample, hindering the separation between them when this number is high. Because of that, the proposed methodology tries to simplify the task of classification, and thus the optimization process. The proposal is intended to construct an ensemble which performs much simpler classification tasks. In order to do so, different decision models are computed, one for separating each class from the remaining ones (avoiding the problem of a great number of classes and aiming at a more balanced classification). The main motivation for this paper could be found in the sentence of Albert Einstein, "Make everything as simple as possible, but not simpler," because the original classification task is simplified, but without forgetting the ordinal ranking information implicit in the data.

Various supervised and disjoint clusters (the term cluster is used to refer to a group of classes) are computed and classified taking into account the natural order of the classes, i.e., a label manipulation procedure is conducted in order to generate multiple hypotheses. In methods that manipulate the target attribute, instead of inducing a single complex classifier, several classifiers are induced with different and usually simpler representations of the target attribute [14]. One example of this is the one-versus-all methodology [25] (previously introduced for SVMs), where a K class classification problem is transformed into K binary classification ones. The one-versus-all paradigm seeks the ith decision function $f_i(\mathbf{x}), i \in \{1, \ldots, K\}$ fulfilling that $f_i(\mathbf{x}) > 0$ when $\mathbf{x}$ belongs to class i, and $f_i(\mathbf{x}) < 0$ when $\mathbf{x}$ belongs to one of the remaining classes. Therefore, f is used as a membership function for choosing the final prediction. The proposal described in this section can be seen as a one-versus-all reformulation for ordinal regression.

In ordinal regression, one-versus-all approach would not compute a fair classification, as the implicit order information would be ignored. For example, for a five-class problem, f_4 will try to distinguish between class 4 and classes $\{1, 2, 3, 5\}$. As class 5 is supposed to be closer to class 4 than to classes $\{1, 2, 3\}$, it might be difficult to separate it from class 4. The proposal tries to separate one class from the previous and the following ones, in such a way that the order among the classes is taken into account (see Fig. 1).

Furthermore, there exists another main issue apart from the exploitation of ordinal ranks by simplifying the classification task. It is well known that the possible ways of combining the outputs of different classifiers in an ensemble depends on what information is obtained from individual members. When dealing with classification algorithms, the most common output for a learning procedure is the label predicted. However, in some cases, there is other information directly extractable from the classifier which may be helpful for improving classification performance, such as predicted probabilities. Threshold methods present the problem of threshold computation which may often be a complex but important issue, as final classification entirely depends on those thresholds. In order to relax and alleviate this kind of errors, probability estimations are carried out by the proposed ensemble methodology.

Let us formally define the method. Given K different classes and corresponding events $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K)$, K different classification problems will be computed by relabeling the data and training the learning algorithm with these relabeled patterns. By doing this, K different models will be obtained.

1) Two of the models (the first one, $i = 1$, and the last one, $i = K$) will compute binary classifications, separating class i from all the others. Standard KDA, SVM or LR will be applied in these cases.
2) The rest of them ($i \in \{2, \ldots, K-1\}$) will be three class classifiers, separating the corresponding class ith from previous ones $(1, \ldots, i-1)$ and subsequent ones $(i+1, \ldots, K)$. Any of the previously presented ordinal algorithms could be used in order to maintain the ordinal rank of the classes (in these cases, the KDO, SVOI, and POM algorithms will be used).

An ensemble set $\mathbb{D}$ will be defined consisting of a combination of K different decision makers, $\mathbb{D} = \{D_1, \ldots, D_K\}$. Each projection will be determined by the set of data to discriminate, as can be seen in Fig. 1 for $K = 5$, where X_i is the set of patterns belonging to class ith.

The training set is defined as $\mathbb{G} = \{\mathbf{G}_1, \ldots, \mathbf{G}_K\}$ for each member of the ensemble, where $\mathbf{G}_i = \{\mathbf{X}_{(j|j<i)}, \mathbf{X}_{(j|j=i)}, \mathbf{X}_{(j|j>i)}\}$. Note that, in the first and last cases, one of the sets to discriminate will be the empty set, as there are no lower and higher ranking classes, respectively. Consequently, the cardinality of $\mathbf{G}_i$ will be $|\mathbf{G}_i| = 3$, for $i \in \{2, \ldots, K-1\}$, and $|\mathbf{G}_i| = 2$ for $i = 1$ and $i = K$.

Clusters grouping different classes will be defined for each decision maker D_i: $\mathcal{C}_{ij}$, $1 \leq i \leq K$. The set of events to classify is defined in the following way: $\{\mathcal{C}_{i1} = (\mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{i-1}), \mathcal{C}_{i2} = \mathcal{C}_i, \mathcal{C}_{i3} = (\mathcal{C}_{i+1} \cup \cdots \cup \mathcal{C}_K)\}$, taking into account that, in the first and the last classification tasks, some of them will be the empty set. These clusters result in different class targets (according to their rank): $S_1 = \{1, 2\}$, $S_i = \{1, 2, 3\}$, $(1 < i < K)$, and $S_K = \{1, 2\}$.

Then, each decision maker (D_i) is determined by the set to discriminate $(\mathbf{G}_i)$, the labels S_i, the computed optimal model (which in this case will be the optimal projection or hyperplane $\mathbf{w}_i$) and the set of thresholds for separating the classes $(\mathbf{b}_i)$. Note that the number of thresholds for the classification corresponds to $|S_i| - 1$.

Although KDA, SVM, and LR have been selected as base methods since they can be easily transformed to predict probabilities, the ensemble could be used with any threshold or probabilistic method. As when using threshold models it is possible to estimate K sets of probability, the first hypothesis is that the true values of $P(\mathcal{C}_i|\mathbf{x}, \mathbb{D})$, i.e., the posterior probability, are the ones most agreed upon by the ensemble.

Although many types of uncertainty exist, probabilistic models fits surprisingly well in most pattern recognition problems [13]. Because of that, this paper tries to construct a classifier by only taking estimated probabilistic information

into account. For each pattern and decision maker i, the probability of belonging to class i will be calculated, along with the probability of belonging to the previous classes and the probability of belonging to the following ones. Then, a methodology for joining all the probabilities is proposed. For that, there are several issues to be addressed.

1) Distributing the probabilities within the cluster: when the specific model for separating class ith from the rest is computed, three (or two) different supervised clusters are formed, one for the classes whose class target is less than i, one for class i and one for the classes whose class target is greater than i. These projections can be used to approximate the probability of belonging to a specific cluster (by using (2) and (3) of the next subsection), where one or more classes are represented. This probability has to be distributed among the different classes included in the cluster to obtain a K-class probability distribution for each decision maker.
2) Combining the probabilities: as in any ensemble, a way has to be selected to combine the decisions of all classifiers (average, product, majority voting, etc).
3) Weighting more prominent classes: after distributing the probabilities, there are classes that are more prominent (for example extreme classes, which appear isolated in two of the projections, see Fig. 1). If a weighting method is not applied, all the patterns will be more likely to be classified in these classes.

A. Obtaining Probability Outputs

An important advantage of threshold methods [4], [10] over other algorithms is that their outputs can be easily transformed into conditional probabilities by analyzing projected patterns and the corresponding thresholds. This is due to the fact that, in high-dimensional feature space, the histogram of each class projected by the discriminant function can be closely approximated by a given distribution. For example, given a pattern $\mathbf{x}$ and a decision maker D_i the probability that this pattern has of belonging to cluster $\mathcal{C}_{ij}$ can be estimated using:

1) the probit function that computes a normal cumulative distribution

$$P(\mathcal{C}_{ij}|\mathbf{x}, D_i) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\mathbf{x}} e^{\frac{-(t-\mu)^2}{2\sigma^2}} dt \tag{2}$$

2) or the logit function that computes the standard logistic sigmoid:

$$P(\mathcal{C}_{ij}|\mathbf{x}, D_i) = \frac{1}{1+e^{-t}} \tag{3}$$

where $i \in \{1, \ldots, K\}$, $j = 1$ or $j \in \{1, 2\}$, $t = \mathbf{w}_i^T\mathbf{x} - b_{ij}$ is the projected pattern, $\mathbf{w}_i^T$ is the ith transposed projection vector, b_{ij} is the corresponding bias for cluster j, and the assumption of $\mu = 0$ and $\sigma = 1$ is made.

Conditional probabilities can be useful, for instance, in applications where the output of a classifier needs to be combined with other information, and it is not only the class assignment that is interesting, but also its probability. Additionally, these probabilities allow us to combine the outputs of K classifiers.

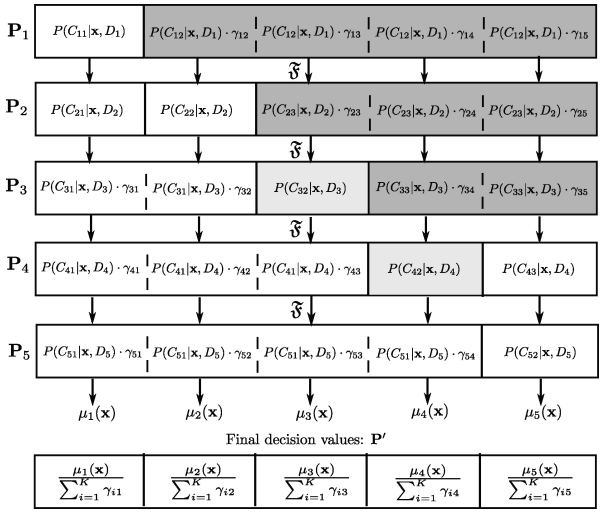


Fig. 2. Example showing the different stages of the procedure. A combination function $\mathfrak{F}$ is used to combine the probability outputs and obtain all $\mu_k(\mathbf{x})$ values.

In this paper, the probit function has been used for estimating the probabilities in the case of the KDA methodologies, since these methods assume an unimodal normal distribution on the data. For LR methods, the logit function was used. On the other hand, as there is no guideline about which method should be used with nonparametric methods, such as SVMs, the logit function has been considered, which has been proved to show good results with this technique [26], [27].

B. Distributing the Probabilities within the Cluster

If the probability that a pattern belongs to a specific cluster is determined by a decision maker D_i, then when the cluster $\mathcal{C}_{ij}$ has only one class, the probability is directly defined but, if there are multiple classes, this probability should be distributed among the classes included in it (as can be seen in Fig.2). One first idea could be simply to ignore all the clusters with more than one class and make use of the independent membership values of the ith single class of each decision maker (after applying the transformations proposed in the previous subsection), in such a way that a vector of decision values $\mathbf{V} = \{P(\mathcal{C}_1|\mathbf{x}, D_1), P(\mathcal{C}_2|\mathbf{x}, D_2), \ldots, P(\mathcal{C}_K|\mathbf{x}, D_K)\} = \{P(\mathcal{C}_{11}|\mathbf{x}, D_1), P(\mathcal{C}_{22}|\mathbf{x}, D_2), \ldots, P(\mathcal{C}_{K2}|\mathbf{x}, D_K)\}$ is computed and the final prediction would be the index of the maximum value of it. Throughout this paper, this methodology is referred to as simple ensemble learning for ordinal regression (SELOR) and has a significant disadvantage: the whole set of probabilities is not being considered.

More complex responses can be obtained if clusters with multiple classes are considered and the corresponding probability is distributed among these classes. One possible way of distributing these probabilities is the following:

$$P(\mathcal{C}_k|\mathbf{x}, D_i) = P(\mathcal{C}_{ij}|\mathbf{x}, D_i)\cdot\gamma_{ik}, \quad \forall(\mathcal{C}_k \in \mathcal{C}_{ij}) \tag{4}$$

with $k \in \{1, \ldots, K\}$, $j \in \{1, 2, 3\}$ or $j = \{1, 2\}$, and taking into account that $\gamma_{ik} = 1$ when $|\mathcal{C}_{ij}| = 1$.

This γ_{ik} weighting parameter could be chosen in many different ways.

1) Equally distributed probabilities: The probability of belonging to class $\mathcal{C}_k$ for a specific decision maker D_i (where $k \in \{1, \ldots, K\}$) is the probability of belonging to the cluster $\mathcal{C}_{ij}$ (taking into account that the patterns $\mathbf{X}_k$ associated to $\mathcal{C}_k$ belongs to cluster $\mathcal{C}_{ij}$) divided by the number of different class targets involved in the cluster, in this case

$$\gamma_{ik} = \frac{1}{|\mathcal{C}_{ij}|}, \quad \forall(\mathcal{C}_k \in \mathcal{C}_{ij}). \tag{5}$$

For the sake of simplicity, this will be the method considered for all the experiments in this paper.

2) Distribution according to the number of patterns in each class: The probability of belonging to class $\mathcal{C}_k$ for a decision maker D_i would be the probability of belonging to cluster $\mathcal{C}_{ij}$ multiplied by the number of patterns in class $\mathcal{C}_k$ with respect to the total involved in the cluster, then

$$\gamma_{ik} = \frac{\sum_{n=1}^{N} I(y_n = \mathcal{C}_k)}{\sum_{n=1}^{N} I(y_n \in \mathcal{C}_{ij})}, \quad \forall(\mathcal{C}_k \in \mathcal{C}_{ij}) \tag{6}$$

where $I(\cdot)$ is defined as the indicator function.

3) Distribution according to the inverse of the number of patterns of each class: The probability of belonging to class $\mathcal{C}_k$ for a decision maker D_i would be, as before, the probability of belonging to cluster $\mathcal{C}_{ij}$ multiplied by the inverse of the number of patterns in class $\mathcal{C}_k$ with respect to the total involved in the cluster, thus

$$\gamma_{ki} = 1 - \frac{\sum_{n=1}^{N} I(y_n = \mathcal{C}_k)}{\sum_{n=1}^{N} I(y_n \in \mathcal{C}_{ij})}, \quad \forall(\mathcal{C}_k \in \mathcal{C}_{ij}).$$

This alternative method could be considered for those unbalanced datasets where there is a special interest in classifying minority classes.

Note that the parameter γ_{ki} is calculated taking into account only training data.

C. Fusion of Probabilities

After applying the method in the above subsection, a matrix $\mathbf{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_K\}$ of probabilities is obtained, where $\mathbf{P}_{i,j} = p_{i,j} = P(\mathcal{C}_j|\mathbf{x}, D_i)$, satisfying that $\sum_{j=1}^{K} p_{i,j} = 1$. Now, all the columns of this matrix are combined to obtain a final decision vector. A nontrainable combiner [13] is considered, i.e., no additional parameters will be tuned, so the ensemble will be ready for classification as soon as the base classifiers are trained. The membership for the jth class is calculated using the jth column of the matrix: $\mu_j(\mathbf{x}) = \mathfrak{F}\left[p_{1,j}(\mathbf{x}), \ldots, p_{K,j}(\mathbf{x})\right]$, where $\mathfrak{F}$ is defined as a combination function. The most commonly used choices for this function are the simple mean

$$\mu_j(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^{K} p_{i,j}(\mathbf{x}) \tag{7}$$

and the product

$$\mu_j(\mathbf{x}) = \prod_{i=1}^{K} p_{i,j}(\mathbf{x}). \tag{8}$$

Pseudocode for the ordinal ensemble proposed

- **Input**: training inputs ($\mathbf{x}_{\mathrm{Tr}}$), training targets ($\mathbf{t}_{\mathrm{Tr}}$), test inputs ($\mathbf{x}_{\mathrm{Ts}}$).
- **Output**: test predicted targets ($\mathbf{t}_{\mathrm{Ts}}$).

for $i = 1$ to K
1. Compute the clusters $\mathbf{G}_i$ from $\mathbf{x}_{\mathrm{Tr}}$ and $\mathbf{t}_{\mathrm{Tr}}$, where $\mathbf{G}_i = \{X_{(j|j<i)}, X_{(j|j=i)}, X_{(j|j>i)}\}$.
2. Train decision maker D_i for $\mathbf{G}_i$: optimal projection $\mathbf{w}_i$ and thresholds ($\mathbf{b}_i$) using either the binary or ordinal algorithm.
3. Project test data.
4. Compute test probabilities of belonging to each cluster.
5. Distribute clustered test probabilities among the classes for obtaining $\mathbf{P}$, equation (4).

end for

Apply the defined $\mathfrak{F}$ function to the matrix $\mathbf{P}$, equations (5) or (6).

Weight each column of $\mathbf{P}$ by using γ_{ij} values ($\mathbf{P}'$), equation (7).

Assign $\mathbf{t}_{\mathrm{Ts}}$ choosing the index of maximum value of each column in the decision vector $\mathbf{P}'$.

Fig. 3. Different steps of the ensemble algorithm.

A theoretical framework is offered for the average and product combiners in [28] based on the Kullback–Leibler divergence, which measures the distance between two probability distributions. These combiners are the two most studied [29], but there is no guideline as to which one is better for a specific problem. In general, the average might be less accurate than the product for some problems, but it is more stable since a small change in a probability makes a bigger impact on the product than on the average.

D. Weighting More Prominent Classes

The distribution of probabilities considered in Section III-B makes some classes receive more attention: for example, in Fig. 1, classes $\mathcal{C}_1$ and $\mathcal{C}_5$ appear isolated in the projections more often than classes $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$, their computed probability being higher (a priori) than that of the other classes. Therefore, a weighting method is used in such a way that

$$P'(\mathcal{C}_i|\mathbf{x}, \mathbb{D}) = \frac{P(\mathcal{C}_i|\mathbf{x}, \mathbb{D})}{\sum_{j=1}^{K} \gamma_{ij}}. \tag{9}$$

E. Further Considerations

In order to clarify all the concepts in previous subsections, a summary of the approach in this paper is given in Fig. 3.

Concerning time complexity, the proposed ensemble will be obviously more time consuming than the base classifier, since it will compute K different models instead of one. However, the models computed will be simpler than the original ones, as the classification problem joins neighbor classes.

IV. Experiments

Several benchmark datasets with different characteristics have been tested in order to validate the methodology proposed. Table I shows the characteristics of these datasets,

TABLE I
CHARACTERISTICS OF THE BENCHMARK DATASETS, ORDERED BY THE NUMBER OF CLASSES

Dataset	No. of Pat.	No. of Attr.	No. of Classes	Class distribution
squash-stored (SS)	52	51	3	(23, 21, 8)
squash-unstored (SU)	52	52	3	(24, 24, 4)
tae (TA)	151	54	3	(49, 50, 52)
newthyroid (NT)	215	5	3	(30, 150, 35)
car (CA)	1728	21	4	(1210, 384, 69, 65)
eucalyptus (EU)	736	91	5	(180, 107, 130, 214, 105)
pyrimx5 (P5)	74	27	5	(15, 15, 15, 15, 14)
machinex5 (M5)	209	7	5	(42, 42, 42, 42, 41)
housingx5 (H5)	506	14	5	(101, 101, 101, 101, 101)
abalonex5 (A5)	4177	11	5	(836, 836, 835, 835, 835)
automobile (AU)	205	71	6	(3, 22, 67, 54, 32, 27)
pyrimx10 (P10)	74	27	10	(8, 8, 8, 8, 7, 7, 7, 7, 7, 7)
machinex10 (M10)	209	7	10	(21, 21, 21, 21, 21, 21, 21, 21, 21, 20)
housingx10 (H10)	506	14	10	(51, 51, 51, 51, 51, 51, 50, 50, 50, 50)
abalonex10 (A10)	4177	11	10	(418, 418, 418, 418, 418, ...418, 418, 417, 417, 417)

where the number of patterns, attributes, classes, and the class distribution (number of patterns per class) can be seen. These publicly available real ordinal classification datasets were extracted from benchmark repositories (UCI [30] and *mldata.org* [31], [32]). Also, some of the ordinal regression benchmark datasets (pyrim, machine, housing, and abalone) provided by Chu *et al.* [12] were considered since they are widely used in the ordinal regression literature [4], [17]. These datasets do not originally represent ordinal classification tasks but regression ones. To turn regression into ordinal classification, the target variable is discretized into K different bins (representing classes, in this case K was assigned to 5 and 10), with equal frequency, as proposed in the previously mentioned works [4], [12], [17].

A. Methods Compared

For an extensive analysis, several methods are compared. The proposed methodologies are applied using both SVM and KDA (and their adaptation to ordinal regression) as base methods. From now on, the methodologies are named as:

1) ensemble learning for ordinal regression using product combiner with SVM and KDA (EPS and EPK);
2) ensemble learning for ordinal regression using average combiner with SVM and KDA (EAS and EAK);
3) simple ensemble learning for ordinal regression with SVM and KDA (SS and SK).

These results have been compared with other state-of-the-art ordinal and nominal methods, such as:

1) *Ordinal Methods:*

a) Kernel discriminant learning for ordinal regression (KDO) [4] and support vector ordinal regression with implicit constraints (SVOI) [17], methods used as base classifiers in the ensemble proposals.
b) Ordinal class classifier using the C4.5 as base classifier (OCC) [3] and ordinal class classifier with specific ordinal weights (OCCW) and SVM [7], both discussed in Section II-B since they are closely related to the proposal.
c) Extreme learning machine for ordinal regression (ELMOR) [33] because the Extreme Learning Machine paradigm has demonstrated good scalability and generalization performance with a faster learning speed when compared to SVM [34].
d) The POM algorithm [2] introduced in Section II-C.

2) *Nominal Methods:*

a) SVM classifier with one-versus-one methodology (SVM1) [35], and one-versus-all formulation (SVMA) [35]. These are the two main approaches for dealing with multiclass problems when using binary classifiers. Both are closely related to the proposal and it seems necessary to verify if they yield similar performances.
b) SVM classifier using a probabilistic reformulation of the one-versus-all paradigm (SVMPA). In this case, the one-versus-all approach is reformulated to estimate probabilities, just like the proposal in this paper. That is, after performing each binary classification, the probabilities for each hypothesis are calculated and later combined by using a combination function (the product, as it has been the one presenting the best results in this experimental section). The purpose of this comparison is to check if the possible improvement of the ELOR method compared to the standard 1VsAll approach is due to the probabilistic component of the proposal. Equally distributed probabilities are considered and a weighting probability method is not necessary, because all the classes receive the same attention.
c) AdaBoostM1 using C4.5 as base classifier (AdaB). This ensemble classifier is one of the most widely used in the machine learning literature, given its proven performance.

KDO and the proposed ensemble variants were implemented using MATLAB, as well as the POM model available through the `mnrfit` function. The authors of SVOI provide a publicly available software,[1] which was considered both for the

[1]http://www.gatsby.ucl.ac.uk∼chuwei/svor.htm

standalone version and the proposed ensemble version. The well-know `libsvm` implementation[2] was considered for all the different versions of the SVM ensembles and for OCCW. The Matlab code for ELM[3] was adapted to implement ELMOR. Finally, the Weka[4] machine learning framework [36] provided the implementations for OCC and AdaB.

B. Evaluated Measures

Several measures can be considered for evaluating ordinal classifiers. The most common ones in machine learning are the mean absolute error (MAE) and the mean zero-one error (MZE) [4], [17], [18], being $MZE = 1 - Acc$, where Acc is the accuracy or correct classification rate. However, as previously said, these measures are not the best option, for example, when measuring performance in the presence of class imbalances [37] and/or when the costs of different errors vary markedly. Because of that, this paper makes use of different kind of measures to evaluate classifier performance.

The MAE is the average deviation in absolute value of the predicted class from the true class [37]: $MAE = \frac{1}{N}\sum_{i=1}^{N} e(\mathbf{x}_i)$, where $e(x_i) = |r(y_i) - r(y_i^*)|$ is the distance between the true and the predicted ranks ($r(y)$ being the rank for a given target y), and, then, MAE values range from 0 to $K-1$ (maximum deviation in number of ranks between two labels).

The average mean absolute error (AMAE) is the mean of the MAE across classes [37]: $AMAE = \frac{1}{K}\sum_{k=1}^{K} MAE_k = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{i=1}^{N_k} e(\mathbf{x}_i)$, where AMAE values range from 0 to $K-1$.

The maximum mean absolute error (MMAE) for all the classes is the MAE value considering only the patterns from the class with the greatest distance between true labels and predicted ones: $MMAE = \max\{MAE_k; k \in \{1, \ldots, K\}\}$, where MAE_k is the MAE value considering only the patterns from the kth class and N_k is the number of pattern in this class. MMAE values range from 0 to $K-1$. This measure was recently proposed [38] and its advantage is that a low MMAE represents a low error for all independently considered classes.

Kendall's τ_b is a statistic used to measure the association between two measured quantities. Specifically, it is a measure of rank correlation $\tau_b = \frac{\sum c_{ij}^* c_{ij}}{\sqrt{\sum c_{ij}^{*2}\sum c_{ij}^2}}$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, N\}$ where c_{ij}^* is $+1$ if y_i^* is greater than (in the ordinal scale) y_j^*, 0 if y_i^* and y_j^* are the same, and -1 if y_i^* is lower than y_j^*, and the same for c_{ij}. τ_b values range from -1 (maximum disagreement between prediction and true label), to 0 (no correlation between them) and to 1 (maximum agreement). One important advantage of this correlation index is that it makes no assumption about the scale of the ranks.

The weighted Kappa (W_k) is a modified version of the Kappa statistic to allow different weights to different levels of aggregation between two variables: $W_k = \frac{p_{o(w)} - p_{e(w)}}{1 - p_{e(w)}}$, with $p_{o(w)} = \frac{1}{n}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij} n_{ij}$, and $p_{e(w)} = \frac{1}{n^2}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij} n_{i\cdot} n_{\cdot j}$, where n_{ij} is the number of times the patterns are predicted by the classifier to be in class j when they really are in class i, $n_{i\cdot} = \sum_{j=1}^{K} n_{ij}$ and $n_{\cdot j} = \sum_{i=1}^{K} n_{ij}$ for $i, j \in \{1, \ldots, K\}$. The weight $w_{ij} = |i - j|$ quantifies the degree of discrepancy between true (y_i) and predicted (y_j^*) categories, and W_k range from -1 to 1.

In this sense, different character measures are used. First, the Acc measure, the most common for classification, reports, in terms of a ratio, how well the classifier works without making any distinction between the classes in the problem. Secondly, the standard MAE measure, well known for ordinal regression problems, considers different misclassification errors. Also, two novel measures are used in order to prove whether the proposal achieves more balanced predictions when the number of patterns is very different for each class. The AMAE metric reports how well all the classes are classified and the MMAE gives information about the worst classified class. Finally, two different statistics are considered, in order to measure the association between prediction and true labeling.

C. Evaluation and Model Selection

Regarding the experimental setup, a holdout stratified technique was applied to divide the datasets 30 times, using 75% of the patterns for training and the remaining 25% for testing. For the regression datasets provided by Chu *et al.* [12] (pyrim, machine, housing, and abalone), the number of random splits was 20 and the number of training and test patterns are the same as those presented in the corresponding works [12], [17]. The partitions were the same for all methods compared and one model was obtained and evaluated (in the test set), for each split. Finally, the results are taken as the mean and standard deviation of the measures over the 30 test sets.

The parameters of each algorithm are chosen using a nested validation with each of the training sets (k-fold method with $k = 5$) and the cross-validation criteria is the MAE since it can be considered the most common one in ordinal regression. The kernel selected for all the algorithms is the Gaussian one, $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)$ where σ is the standard deviation.

For every tested kernel method (KDA and SVM methods), the kernel width was selected within these values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, as the cost parameter associated with SVM methods. The parameter u for avoiding singularity (for the methods based on KDA) was selected within $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and the C parameter for the KDO was selected within the following ones $\{10^{-1}, 10^{0}, 10^{1}\}$.

D. Results

This section presents three different types of experiments. First, a synthetic dataset is designed in order to show the advantages of the proposal graphically when comparing it with the one-versus-all standard formulation. Second, the results obtained are compared for the 15 datasets previously presented, with the 6 ensemble methodologies proposed and 10 state-of-the-art algorithms, using a set of 6 different selected measures. Finally, a different set of experiments with large-scale ordinal datasets is performed to analyze the potential scalability and interpretability of the proposed method.

1) *Graphical Representation of the Proposal:* In this subsection, a new synthetic dataset has been designed in order to show the main differences and advantages when comparing

[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[3]http://www.ntu.edu.sg/home/egbhuang/

[4]http://www.cs.waikato.ac.nz/ml/weka/

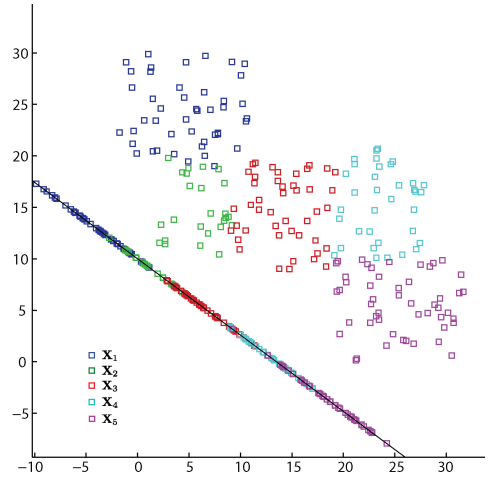


Fig. 4. Synthetic dataset and the optimal projection computed by linear discriminant analysis for ordinal regression [4].

the ordinal version of the algorithms and the one-versus-all standard proposal. The graphical representation of the dataset can be seen in Figs. 4 and 5.

Fig. 4 shows the ordinal projection computed and the projected patterns for the dataset. Linear discriminant analysis for ordinal regression has been used for this (without using the kernel trick) to allow the representation of the results, due to the fact that the kernel trick would classify the dataset structure perfectly. Taking into account the final projection, it can be observed that classes 2, 3, and 4 are not very well classified since they present some overlapping on the projection.

On the other hand, Fig. 5 shows various projections computed and the patterns projected by the proposed procedure and the one-versus-all paradigm (also using linear discriminant analysis). The computed projections for the first and last classes are seen to be the same as in the previous case, since the classification tasks are the same. But in this case, the projection for the rest of the classes allows better separation since some order among the classes is supposed. For example, $\mathbf{w}_3$ allows a clearer separation for the classes than the computed $\mathbf{w}_3$ in the one-versus-all formulation, where the classes $\{1, 2, 3\}$ are mixed in the projection. Each single class is seen to be well classified in at least one model, and also the classes are ordered in the projection so that some information is implicit in the model.

2) *Experimental Results:* The algorithms compared here have been run and optimized under the same conditions and using the same parameter cross-validation. First, the different ensemble proposals and their base algorithms are compared, and then the rest of the state-of-the-art methods are considered.

Table II shows the mean ranking for the proposals and the base methodologies for all 15 datasets, taking into account 6 different measures, which may help the reader to evaluate the value of the proposal. This table only considers the mean ranking (over all the datasets) obtained for each method and each metric. In this case (where eight algorithms were compared), a ranking of 1 is assigned to the best method for a given dataset, and a ranking of 8 to the one which provides the worst performance. In this table and all the following ones,

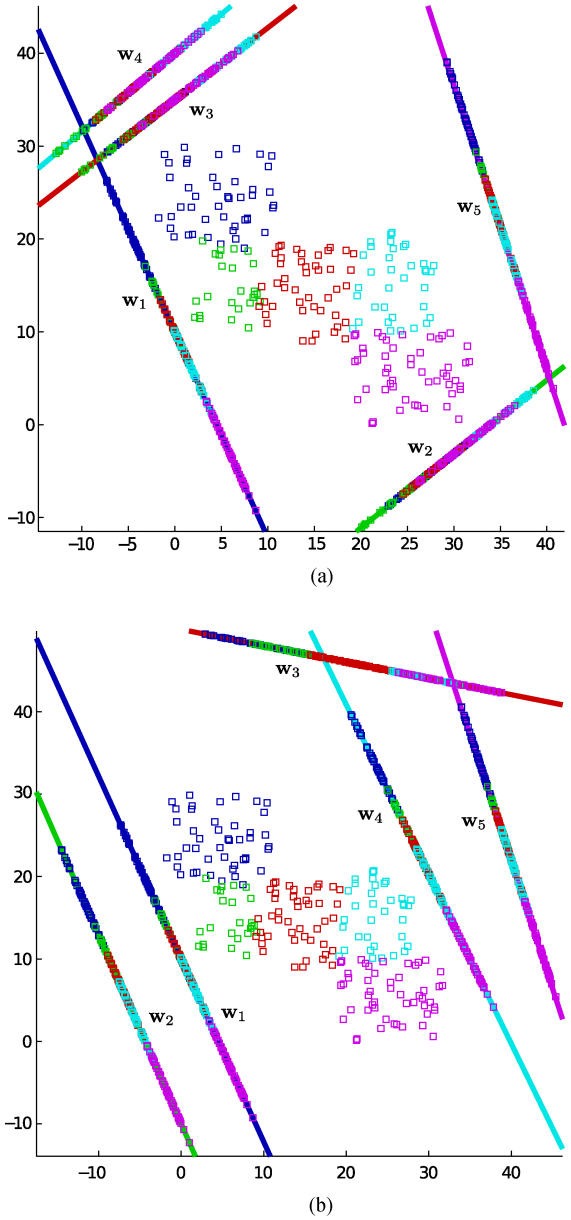


Fig. 5. Graphical representation of the different projections computed for the synthetic dataset using linear discriminant analysis. (a) Projections computed by the one-versus-all formulation. (b) Projections computed by the ordinal ensemble methodology proposed.

the best method is in bold face and the second one in italics. The mean ranking considering all the metrics has also been included in the table as a summary. In almost all cases, the ensemble achieves better results than the initial algorithms. Specifically, it can be seen that the best results or the second best results for almost all the metrics tested are achieved by applying the EPS proposal. The complete tables of results showing the means and standard deviations for all benchmark datasets and metrics are not included in this paper for the sake of simplicity and readability, but they can be found on a public webpage.[5]

[5]http://www.uco.es/grupos/ayrna/elor2013

TABLE II
MEAN RANKINGS OF THE 15 DATASETS CONSIDERED FOR THE ENSEMBLE METHODOLOGIES PROPOSED AND THE BASE ALGORITHMS USED

	Method							
Measure	KDO	EPK	EAK	SK	SVOI	EPS	EAS	SS
Acc	6.73	3.87	4.80	5.77	3.87	**2.07**	*3.50*	5.40
MAE	6.27	4.70	4.60	6.67	3.30	*2.67*	**2.60**	5.20
AMAE	6.27	4.70	4.37	6.27	*3.00*	**2.53**	3.33	5.53
MMAE	5.27	4.83	3.70	5.67	*3.53*	**3.40**	4.00	5.60
τ_b	5.47	3.97	4.57	7.00	3.27	**2.87**	*3.07*	5.80
W_k	5.73	3.57	5.23	6.87	*3.47*	**2.00**	3.60	5.53
Average	5.96	4.27	4.54	6.37	3.41	**2.59**	*3.35*	5.51

To quantify whether a statistical difference exists among the algorithms compared in Table II, a procedure is employed to compare multiple classifiers in multiple datasets [39]. First of all, a Friedman's nonparametric test with a significance level of $\alpha = 0.05$ has been carried out to determine the statistical significance of the differences in the mean ranking results for each measure selected. The test rejected the null hypothesis that all algorithms perform similarly when $\alpha = 0.05$ for all the selected metrics, stating then that the differences in mean rankings of *Acc*, MAE, AMAE, MMAE, Kendall's τ_b and W_k are statistically significant. Specifically, the confidence interval for this number of datasets and algorithms is $C_0 = (0, F_{(\alpha=0.05)} = 2.10)$, and the corresponding F-value for each metric was $7.95 \notin C_0$, $9.30 \notin C_0$, $7.65 \notin C_0$, $2.47 \notin C_0$, $8.11 \notin C_0$ and $10.22 \notin C_0$ for *Acc*, MAE, AMAE, MMAE, Kendall's τ_b and W_k, respectively.

On the basis of this rejection, the Nemenyi post-hoc test is used to compare all classifiers to one another. This test considers that the performance of any two classifiers is deemed significantly different if their mean ranks differ by at least the critical difference (CD), which depends on the number of datasets and methods. 5% significance confidence was considered ($\alpha = 0.05$) to obtain this CD and the results can be observed in Fig. 6, which shows CD diagrams as proposed in [39]. Each method is represented as a point on a ranking scale, corresponding to its mean ranking performance. CD segments are included to measure the separation needed between methods in order to assess statistical differences. Red lines group algorithms for which statistically significant, different mean ranking performance cannot be assessed.

From the results of the statistical tests and from the tables, several conclusions can be drawn: first, one could notice by analyzing mean rankings that the techniques based on SVMs present a clearly better performance than the ones based on KDA, and the ensemble procedure based on SVM usually outperforms the results obtained by the ensemble based on KDA, independently of the combiner or metric used. Secondly, no significant differences can be observed by analyzing different probability combiners, although the great majority of the results show better performance using the product combiner. Also, the methodology SELOR (SK and SS) cannot be considered to be a good approach since its performance is worse than that of the base algorithms

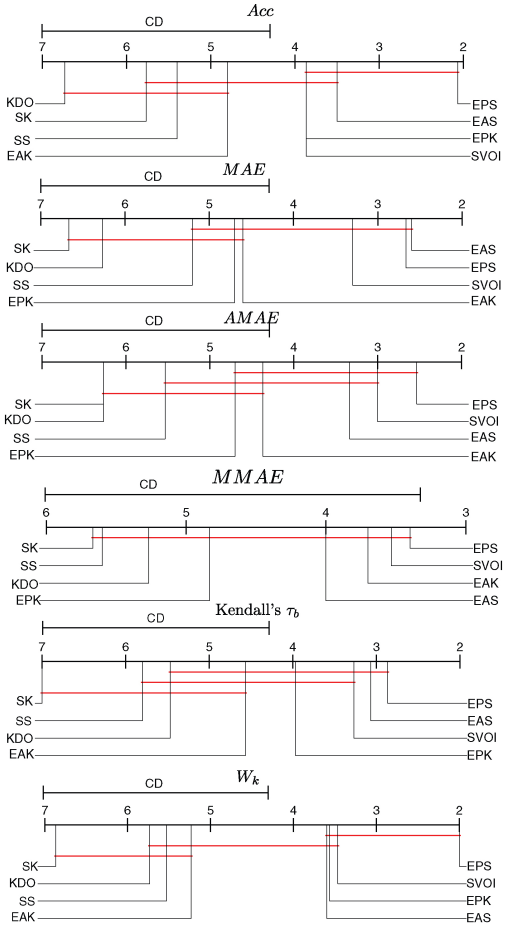


Fig. 6. Results and ranking of the Nemenyi statistical test for proposals and base methods.

in many cases. Thus, it has been shown that considering all the probability information, performance can be significantly improved. Last but not least, the ensemble procedure seems to be a good approach to tackle ordinal regression since it leads to an improvement in the results obtained by several algorithms of the state-of-the-art (as the base classifiers used: KDO and SVOI) taking different measures into account. This can be easily seen by analyzing the Nemenyi post-hoc figures.

To complete this section, a table similar to the previous one but containing the mean rankings for the rest of the state-of-the-art algorithms is shown in Table III. The EPS proposal is also included in this table, since, as stated before, it could be considered the proposal with the best performance. This table shows that the EPS procedure seems to be competitive for all measures (both ordinal and nominal), since it always obtains the best mean ranking. The second best method is OCCW, with the second position for all measures.

Table III shows that the EPS methodology is the best one in performance for all 6 metrics, improving the performances of 4 different ordinal classifiers and 4 nominal ones, and achieving a considerable balance between *Acc*, ordinal measures, those appropriate for imbalanced datasets, and correlation ones.

In this case, the nonparametric Friedman's test with a significance level of $\alpha = 0.05$ was also applied to the

TABLE III
MEAN RANKINGS OF THE 15 DATASETS FOR THE SELECTED ENSEMBLE METHODOLOGY AND OTHER STATE-OF-THE-ART METHODS

	Method								
Measure	EPS	OCCW	OCC	ELMOR	POM	SVM1	SVMA	SVMPA	AdaB.
Acc	**2.03**	*2.60*	6.67	5.20	6.23	3.50	5.40	7.43	5.93
MAE	**1.63**	*3.07*	6.10	4.67	5.93	4.57	6.00	7.30	5.73
AMAE	**1.67**	*3.60*	6.20	4.80	5.40	4.70	5.90	7.00	5.73
MMAE	**1.67**	*4.00*	6.27	4.27	4.93	5.27	6.13	7.00	5.47
τ_b	**1.60**	*2.87*	6.60	4.73	5.20	4.53	6.00	7.33	6.13
W_k	**1.40**	*3.07*	6.73	5.27	5.27	4.27	5.73	7.33	5.93
Average	**1.67**	*3.20*	6.43	4.82	5.49	4.47	5.86	7.23	5.82

mean rankings for each measure. The test rejected the null-hypothesis that all algorithms perform similarly when $\alpha = 0.05$ for all the selected metrics, stating then that the differences in the mean ranking of *Acc*, MAE, AMAE, MMAE, Kendall's τ_b and W_k are statistically significant. Specifically, the confidence interval for this number of datasets and algorithms is $C_0 = (0, F_{(\alpha=0.05)} = 2.02)$, and the corresponding F-value for each metric was $12.33 \notin C_0$, $9.52 \notin C_0$, $7.08 \notin C_0$, $6.91 \notin C_0$, $11.24 \notin C_0$ and $11.63 \notin C_0$ for *Acc*, MAE, AMAE, MMAE, Kendall's τ_b and W_k, respectively.

It is well known that the Nemenyi approach comparing all classifiers to one another in a post-hoc test is not as sensitive as the approach comparing all classifiers to a given classifier (known as a control method) [39]. The Holm test performs this latter type of comparison, only considering the comparison between the control method and all the alternatives, and sequentially testing the hypotheses ranked by their significance. The ordered p-values will be denoted by $p_1 \leq p_2 \leq \cdots \leq p_{k-1}$, where k is the number of comparisons made. This step-down procedure compares p_i with a corrected version of the level of significance $\alpha/(k-1)$, starting with the most significant p-value (p_1). If p_i is below the corrected α, the null hypothesis is rejected and the next comparison is performed. When a certain null hypothesis cannot be rejected, all the remaining ones are also retained. The results of this test (corrected α values and p-values) for all the measures are included in Table IV, where EPS is used as the control method.

This table shows that the EPS presents statistically significant differences for $\alpha = 0.05$ for almost all measures with respect to almost all methods, except for SVM1 (when using *Acc*) and OCCW (when using some of the metrics). No statistically significant differences could be assessed when comparing EPS to SVM1 for *Acc*, which is, in fact, a nominal method not designed to deal with ordinal problems. Furthermore, it can be seen that the proposal presents significant statistical differences for $\alpha = 0.05$ and the MMAE metric with respect to the OCCW methodology, which could be considered the procedure most similar to the one designed in this paper, and that the differences for AMAE and W_k are also significant for $\alpha = 0.10$. In any case, it is important to remember that the mean rankings are always the best for EPS.

From these results, several conclusions can be drawn: first, as said before, it has been proven that an ordinal regression point of view is needed when dealing with some given order among categories, because, although a nominal algorithm may perform well when taking into account, for example, the measure of accuracy, it may fail when taking into account other ordinal measures. Secondly, as statistically significant differences exist for all the metrics selected when taking into account the different one-versus-all proposals (the nominal proposal for reformulating the SVM paradigm and the proposal in this paper), ELOR seems to present clear advantages over the one-versus-all nominal paradigm, when tackling ordinal classification. Finally, it can be concluded that the combination of single classifiers, aiming at a more accurate classification decision at the expense of increased complexity, seems to be a good idea in this case, since it improves the performance of other state-of-the-art methodologies significantly.

3) *Large-Scale Datasets and Interpretability:* Once the performance of the proposed method has been extensively validated making statistical comparisons to other state-of-the-art methodologies for different measures and datasets, there are some unanswered issues such as the scalability of the algorithm or its possible interpretability, which is the main aim of this subsection. However, these issues are more related to the choice of the base algorithm for the ensemble because it will obviously determine if the algorithm could be used with large-scale datasets or for model interpretability purposes. The complexity of the kernel methods previously used as base methodologies for the ensemble depends directly on the number of training patterns [4] and their interpretability is difficult. Because of this reason, a simpler and more interpretable method is used for the following experiments. This method does not present parameters to optimize and it is also designed for ordinal regression. It is a linear model, leading generally to a lower performance (see Table III of this paper or other studies in the ordinal classification literature [18], [19]). However, it provides us with a probabilistic output, a simpler model and better interpretability. The method used is the POM algorithm [2] which was used for comparison purposes in the previous experimental subsection. Moreover, standard binary LR is used for the binary decompositions. This methodology, can be considered as interpretable in the sense that it could give us clues about the importance of each attribute for modeling the dependent variable.

For the experiments, two real ordinal datasets have been used. First, the Happiness dataset was extracted from the European Social Survey[6] considering year 2010 and 26 countries. It represents the complex problem of predicting the individual happiness by using certain characteristics, beliefs and life circumstances in a Likert scale (examples of some input variables are: the health of the person, if he or she has anyone to discuss personal matters, whether he or she takes part in social activities, etc). We selected 13 attributes and considered five classes. The dataset was composed of $41,472$ instances (missing values were removed for simplicity). For more information of this dataset see the webpage associated to this paper[5]. Second, the SpanishFleet dataset was obtained from the Fleet Register On the Net considering year 2012 and the whole Spanish fleet to predict the commitment to sustainability of the Spanish vessels, using a categorization

[6]http://ess.nsd.uib.no/

TABLE IV

RESULTS OF THE HOLM PROCEDURE USING EPS AS THE CONTROL METHOD WHEN COMPARED TO OTHER STATE-OF-THE-ART METHODS: CORRECTED α VALUES, COMPARED METHOD AND p-VALUES, ALL OF THEM ORDERED BY THE NUMBER OF COMPARISON (i)

			Acc		MAE		AMAE	
i	$\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i	Method	p_i
1	0.0063	0.0125	SVMPA	0.0000$_\bullet$	SVMPA	0.0000$_\bullet$	SVMPA	0.0000$_\bullet$
2	0.0071	0.0143	OCC	0.0000$_\bullet$	OCC	0.0000$_\bullet$	OCC	0.0000$_\bullet$
3	0.0083	0.0167	POM	0.0000$_\bullet$	SVMA	0.0000$_\bullet$	SVMA	0.0000$_\bullet$
4	0.0100	0.0200	AdaB.	0.0001$_\bullet$	POM	0.0000$_\bullet$	AdaB.	0.0001$_\bullet$
5	0.0125	0.0250	SVMA	0.0008$_\bullet$	AdaB.	0.0000$_\bullet$	POM	0.0002$_\bullet$
6	0.0167	0.0333	ELMOR	0.0015$_\bullet$	ELMOR	0.0024$_\bullet$	ELMOR	0.0017$_\bullet$
7	0.0250	0.0500	SVM1	0.1425	SVM1	0.0034$_\bullet$	SVM1	0.0024$_\bullet$
8	0.0500	0.1000	OCCW	0.5709	OCCW	0.1518	OCCW	0.0532$_\circ$
			MMAE		τ_b		W_k	
i	$\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i	Method	p_i
1	0.0063	0.0125	SVMPA	0.0000$_\bullet$	SVMPA	0.0000$_\bullet$	SVMPA	0.0000$_\bullet$
2	0.0071	0.0143	OCC	0.0000$_\bullet$	OCC	0.0000$_\bullet$	OCC	0.0000$_\bullet$
3	0.0083	0.0167	SVMA	0.0000$_\bullet$	AdaB.	0.0000$_\bullet$	AdaB.	0.0000$_\bullet$
4	0.0100	0.0200	AdaB.	0.0001$_\bullet$	SVMA	0.0000$_\bullet$	SVMA	0.0000$_\bullet$
5	0.0125	0.0250	SVM1	0.0003$_\bullet$	POM	0.0003$_\bullet$	POM	0.0001$_\bullet$
6	0.0167	0.0333	POM	0.0011$_\bullet$	ELMOR	0.0017$_\bullet$	ELMOR	0.0001$_\bullet$
7	0.0250	0.0500	ELMOR	0.0093$_\bullet$	SVM1	0.0034$_\bullet$	SVM1	0.0042$_\bullet$
8	0.0500	0.1000	OCCW	0.0196$_\bullet$	OCCW	0.2053	OCCW	0.0956$_\circ$

$\bullet$: Statistical difference with $\alpha = 0.05$
$\circ$: Statistical difference with $\alpha = 0.10$

of the overexploitation of the gears employed provided by the Food and Agriculture Organization of the United Nations. This dataset was composed of $10,460$ instances, six attributes and ten classes. For more information of this dataset see [40].

Concerning the experiments on these datasets, the same aforementioned experimental design was used (i.e., 30 random repetitions of a stratified holdout, with 75% for training and 25% for the test set). To analyze the scalability of the algorithms, the complete time in seconds for executing each algorithm is also included in the results (note that the same machine architecture was used). The methods tested are: 1) the POM algorithm (which was previously presented), 2) ordinal class classifier using POM as base classifier (OCCP), and 3) ensemble learning for ordinal regression using product combiner and the POM algorithm as base classifier (EPP). We considered OCCP because it is a decomposition method which can be said to follow the same philosophy of EPP but using binary classifiers.

The results of these experiments can be seen in Table V. From these results, it can be seen that the proposed method outperforms in all the metrics the base classifier and, in most of the cases, the ordinal binary decomposition method (OCCP), thus providing more robust results. Furthermore, although both classification problems are complex because of the variable to predict, the obtained results are very promising (e.g., in MAE and AMAE). With regard to the execution time, the computational complexity of the methodology is affordable, even for large-scale problems. Furthermore, as it can be seen in the experiments (comparing the time obtained in both datasets), the time complexity of the algorithm depends to a greater extent on the number of classes (because it determines the number of decompositions to perform) rather than on the number of samples.

TABLE V

MEAN TEST VALUES FOR THE DIFFERENT METHODS CONSIDERED

	Happiness		
Metrics	POM	OCCP	EPP
Acc	60.78 ± 0.15	$\mathit{63.44 \pm 0.26}$	$\mathbf{63.73 \pm 0.25}$
MAE	0.449 ± 0.002	$\mathit{0.402 \pm 0.003}$	$\mathbf{0.397 \pm 0.003}$
AMAE	1.259 ± 0.011	$\mathit{1.028 \pm 0.016}$	$\mathbf{1.002 \pm 0.014}$
MMAE	2.580 ± 0.051	$\mathit{1.953 \pm 0.081}$	$\mathbf{1.950 \pm 0.075}$
τ_b	0.232 ± 0.007	$\mathit{0.350 \pm 0.007}$	$\mathbf{0.375 \pm 0.006}$
W_k	0.088 ± 0.005	$\mathit{0.256 \pm 0.006}$	$\mathbf{0.293 \pm 0.005}$
Time	$\mathbf{23.41 \pm 4.34}$	$\mathit{32.49 \pm 0.50}$	49.00 ± 0.47
	SpanishFleet		
Metrics	POM	OCC(POM)	EPP
Acc	83.33 ± 0.52	$\mathbf{86.62 \pm 0.39}$	$\mathit{85.87 \pm 0.28}$
MAE	0.443 ± 0.012	$\mathit{0.406 \pm 0.015}$	$\mathbf{0.388 \pm 0.010}$
AMAE	$\mathit{2.104 \pm 0.048}$	2.200 ± 0.117	$\mathbf{1.943 \pm 0.062}$
MMAE	6.880 ± 0.116	$\mathbf{5.996 \pm 0.370}$	$\mathit{6.594 \pm 0.151}$
τ_b	$\mathit{0.611 \pm 0.013}$	0.602 ± 0.021	$\mathbf{0.631 \pm 0.017}$
W_k	0.620 ± 0.012	$\mathit{0.665 \pm 0.013}$	$\mathbf{0.678 \pm 0.008}$
Time	$\mathbf{44.21 \pm 2.24}$	$\mathit{173.52 \pm 1.33}$	225.18 ± 2.11

Concerning interpretability, the decomposition proposed provides us with additional information in the sense that one model for differentiating each class from the previous and following classes is computed. Therefore, instead of being provided with a model for tackling the whole learning problem, we obtain a model for discriminating each class and we could analyze independently the variables most determining.

To better visualize the interpretability of the model, let us analyze an example with the Happiness dataset. The best model (in this case the one performing better in terms of MAE for EPP) has been selected for the analysis. This model can be seen in Table VI. Note that both D_1 and D_5 are binary classifiers with a single threshold. The most important variables for modeling the labeling are the ones with higher

TABLE VI
BEST SET OF MODELS D_i, $1 \leq i \leq 5$, OBTAINED BY THE PROPOSED ORDINAL ENSEMBLE USING THE POM ALGORITHM AS BASE METHOD

	D_1	D_2	D_3	D_4	D_5
$\mathbf{w}_1$	0.3172	0.1249	0.1065	0.0427	−0.1027
$\mathbf{w}_2$	0.0194	0.1485	0.1837	0.1829	0.1323
$\mathbf{w}_3$	0.2566	0.2196	0.1348	0.1175	0.0940
$\mathbf{w}_4$	1.0764	0.6465	0.5346	0.4028	0.1987
$\mathbf{w}_5$	0.1906	0.0614	0.0799	0.0551	−0.0239
$\mathbf{w}_6$	0.0873	0.2394	0.2218	0.1990	0.1711
$\mathbf{w}_7$	−0.2139	−0.1879	−0.2126	−0.2018	−0.0667
$\mathbf{w}_8$	−0.0257	0.0983	0.0774	0.0811	0.0954
$\mathbf{w}_9$	−0.0651	−0.1359	−0.1667	−0.1457	−0.0851
$\mathbf{w}_{10}$	−0.6461	−0.5314	−0.4974	−0.4229	−0.2605
$\mathbf{w}_{11}$	0.2036	0.1255	0.0849	0.0584	0.0083
$\mathbf{w}_{12}$	0.3355	0.2957	0.2535	0.1768	0.0819
$\mathbf{w}_{13}$	0.0358	−0.1183	−0.1664	−0.1968	−0.3089
b_1	−6.6354	−6.0191	−3.4551	−0.9498	2.5098
b_2	–	–3.6051	−1.0537	2.8159	–

Meaning of each variable can be found in the website associated to this paper[5].

$|\mathbf{w}_i|$ value, for example, it can be seen that $\mathbf{x}_4$ (satisfaction with present state of economy in country) presents a high impact on the variable to predict and so does $\mathbf{x}_{10}$ (the subjective health of the person). One should note that although the sign of $\mathbf{w}_i$ could also be used for an interpretability analysis, it could depend on the variable coding (in the case of the subjective health the variable is encoded from very good health to very bad health, thus this variable is negatively correlated with the label). Furthermore, it can be seen that variables important for different models are not so determining for others (analyze the case of $\mathbf{x}_1$, $\mathbf{x}_2$ or $\mathbf{x}_{13}$). Besides, as part of the model analysis, it can be said that having someone to discuss personal matters ($\mathbf{x}_7$) makes you happier (note that the "yes" has been encoded as 0 and "no" as 1).

As a final remark, if we order the variables taking into account their importance for each model (as said, the $|\mathbf{w}_i|$ value), it can be observed that some variables have almost no influence for discriminating certain classes (see Table VII). For example: being member of a group discriminated in your country or not ($\mathbf{x}_{11}$) is an influential variable for determining if you are extremely unhappy, but not for determining if you are extremely happy (it is at the last position). On the contrary, thinking that is important to help people and care for others well being ($\mathbf{x}_{13}$) is indeed a determining variable for the happiest (it is at the first position).

TABLE VII
RANKING OF VARIABLES FOR THE DIFFERENT MODELS

D_1	D_2	D_3	D_4	D_5
$\mathbf{x}_4$	$\mathbf{x}_4$	$\mathbf{x}_4$	$\mathbf{x}_{10}$	$\mathbf{x}_{13}$
$\mathbf{x}_{10}$	$\mathbf{x}_{10}$	$\mathbf{x}_{10}$	$\mathbf{x}_4$	$\mathbf{x}_{10}$
$\mathbf{x}_{12}$	$\mathbf{x}_{12}$	$\mathbf{x}_{12}$	$\mathbf{x}_7$	$\mathbf{x}_4$
$\mathbf{x}_1$	$\mathbf{x}_6$	$\mathbf{x}_6$	$\mathbf{x}_6$	$\mathbf{x}_6$
$\mathbf{x}_3$	$\mathbf{x}_3$	$\mathbf{x}_7$	$\mathbf{x}_{13}$	$\mathbf{x}_2$
$\mathbf{x}_7$	$\mathbf{x}_7$	$\mathbf{x}_2$	$\mathbf{x}_2$	$\mathbf{x}_1$
$\mathbf{x}_{11}$	$\mathbf{x}_2$	$\mathbf{x}_9$	$\mathbf{x}_{12}$	$\mathbf{x}_8$
$\mathbf{x}_5$	$\mathbf{x}_9$	$\mathbf{x}_{13}$	$\mathbf{x}_9$	$\mathbf{x}_3$
$\mathbf{x}_6$	$\mathbf{x}_{11}$	$\mathbf{x}_3$	$\mathbf{x}_3$	$\mathbf{x}_9$
$\mathbf{x}_9$	$\mathbf{x}_1$	$\mathbf{x}_1$	$\mathbf{x}_8$	$\mathbf{x}_{12}$
$\mathbf{x}_{13}$	$\mathbf{x}_{13}$	$\mathbf{x}_{11}$	$\mathbf{x}_{11}$	$\mathbf{x}_7$
$\mathbf{x}_8$	$\mathbf{x}_8$	$\mathbf{x}_5$	$\mathbf{x}_5$	$\mathbf{x}_5$
$\mathbf{x}_2$	$\mathbf{x}_5$	$\mathbf{x}_8$	$\mathbf{x}_1$	$\mathbf{x}_{11}$

V. CONCLUSION AND FUTURE WORK

The methodology here proposed was based on the computation of different classification tasks, by performing a relabeling process which took ordinal data information into account. The relabeled data was then used for training the learning algorithm. In that sense, the proposal can be seen as a reformulation of the one-versus-all idea to tackle ordinal regression, as each single model was computed to differentiate each class from the remaining ones taking ordinal ranks into account. Threshold models were used as the base classifier because they are able to include the order information of these groups of classes and their natural projection capabilities facilitate the computation of probability estimations. For the prediction phase, two of the most widely studied combiners in the ensemble literature were used, the product and the average.

The proposal was tested with 15 benchmark datasets and it was found to be competitive when compared to the base classifiers and to other state-of-the-art methods. Statistical tests were applied to assess these conclusions.

Additionally, the superiority of the proposal for the one-versus-all standard paradigm was confirmed when dealing with ordinal regression. Although multiclass imbalance problems pose important difficulties for machine learning algorithms [41], this approach seems to achieve not only good global performance, but also good error rates for all classes independently, given the good MMAE performance obtained.

Moreover, the proposal was seen to be scalable (although this is an issue related to the base methodology, it was seen to provide a reasonable time complexity compared to the base method) and interpretable (in the sense that the most determining features for modeling each class can be extracted because it was based on a decomposition strategy).

Unlike discriminant analysis (where a normal distribution could be assumed), there is no guideline about the probability distribution to use when working with nonparametric approaches, such as SVMs. In fact, several studies have been performed in order to reformulate SVMs to allow probabilistic outputs [26], [27] making use of a maximum-likelihood estimator for adjusting the probability distribution to the projected patterns. This idea might be used as well in this paper in order to compute fairer probabilities for the SVM methodologies.

Finally, the ensemble procedure could be tested with other ordinal base classifiers, also based on SVMs or discriminant analysis such as those proposed in [24] and [8].

REFERENCES

[1] M. Pérez-Ortiz, P. A. Gutiérrez, C. Hervás-Martínez, J. Briceño, and M. de la Mata, "An ensemble approach for ordinal threshold models applied to liver transplantation," in *Proc. IJCNN*, 2012, pp. 2795–2802.
[2] P. McCullagh, "Regression models for ordinal data," *J. Royal Stat. Soc.*, vol. 42, no. 2, pp. 109–142, 1980.

[3] E. Frank and M. Hall, "A simple approach to ordinal classification," in *Proc. 12th Eur. Conf. Mach. Learning*, 2001, pp. 145–156.
[4] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li, "Kernel discriminant learning for ordinal regression," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 6, pp. 906–910, Jun. 2010.
[5] S. Kramer, G. Widmer, B. Pfahringer, and M. de Groeve, "Prediction of ordinal classes using regression trees," in *Proc. 12th ISMIS*, LNCS 1932/2010. Oct. 2000, pp. 665–674.
[6] S. B. Kotsiantis and P. E. Pintelas, "A cost sensitive technique for ordinal classification problems," in *Proc. 3rd Hellenic Conf. Artif. Intell. (SETN)*, LNCS 3025. May 2004, pp. 220–229.
[7] W. Waegeman and L. Boullart, "An ensemble of weighted support vector machines for ordinal regression," *Int. J. Comput. Syst. Sci. Eng.*, vol. 3, no. 1, pp. 1–7, 2009.
[8] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Advances in Neural Information Processing Systems*, vol. 19, B. Schölkopf, J. Platt, and T. Hoffman, Eds. 2007, pp. 865–872.
[9] J. S. Cardoso and J. F. P. da Costa, "Learning to classify ordinal data: The data replication method," *J. Mach. Learning Res.*, vol. 8, pp. 1393–1429, Jan. 2007.
[10] A. Shashua and A. Levin, "Ranking with large margin principle: Two approaches," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA, USA: MIT Press, 2003, pp. 937–944.
[11] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed., Monographs on Statistics and Applied Probability. London, U.K.: Chapman & Hall/CRC, 1989.
[12] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *J. Mach. Learning Res.*, vol. 6, no. 1, pp. 1019–1041, 2005.
[13] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. New York, NY, USA: Wiley-Interscience, 2004.
[14] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, nos. 1–2, pp. 1–39, 2009.
[15] S. Mika, "Fisher discriminant analysis with kernels," Ph.D. dissertation, Univ. Technology, Berlin, Germany, Dec. 2002.
[16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge, U.K.: Cambridge University, 2000.
[17] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Comput.*, vol. 19, no. 3, pp. 792–815, Mar. 2007.
[18] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez, "An experimental study of different ordinal regression methods and measures," in *Proc. 7th Int. Conf. HAIS*, LNCS 7209. 2012, pp. 296–307.
[19] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tino, and C. Hervás-Martínez, "Exploitation of pairwise class distances for ordinal classification," *Neural Comput.*, vol. 25, pp. 1–36, 2013.
[20] S. Menard. (2009). *Logistic Regression: From Introductory to Advanced Concepts and Applications*, New York, NY, USA: Sage Publications [Online]. Available: http://books.google.es/books?id=KuRWdnoe4WUC
[21] B. Fang, Y. Y. Tang, Z. Shang, and B. Xu, "Generalized discriminant analysis: A matrix exponential approach," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 40, no. 1, pp. 186–197, Feb. 2010.
[22] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[23] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *Proc. Int. Conf. Artif. Neural Netw.*, 1999, pp. 97–102.
[24] A. Shashua and A. Levin, "Advances in neural information processing systems," in *Ranking with Large Margin Principle: Two Approaches*, vol. 15, Cambridge, MA, USA: MIT Press, 2003, pp. 937–944.
[25] R. Anand, K. Mehrotra, C. Mohan, and S. Ranka, "Efficient classification for multiclass problems using modular neural networks," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 117–124, Jan. 1995.
[26] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambrdige, MA, USA: MIT Press, 1999, pp. 61–74.
[27] V. Franc, A. Zien, and B. Schölkopf, "Support vector machines as probabilistic models," in *Proc. ICML*, 2011, pp. 665–672.
[28] D. J. Miller and L. Yan, "Ensemble classification by critic-driven combining," in *Proc. IEEE ICASSP—Volume 02*, Mar. 1999, pp. 1029–1032.
[29] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.
[30] A. Asuncion and D. Newman. (2007). *UCI machine learning repository* [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html
[31] PASCAL (2011). *Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository* [Online]. Available: http://mldata.org/
[32] D. S. Sonnenburg. (2011). *Machine Learning Data Set Repository* [Online]. Available: http://mldata.org/
[33] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, and X. Wang, "Ordinal extreme learning machine," *Neurocomputation*, vol. 74, nos. 1–3, pp. 447–456, Dec. 2010.
[34] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[35] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.
[36] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *ACM SIGKDD Explor. Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
[37] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Proc. 9th Int. Conf. ISDA*, Nov.–Dec. 2009, pp. 283–287.
[38] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "A preliminary study of ordinal metrics to guide a multi-objective evolutionary algorithm," in *Proc. 11th Int. Conf. ISDA*, Nov. 2011, pp. 743–747.
[39] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, no. 1, pp. 1–30, 2006.
[40] M. Pérez-Ortiz, R. Colmenarejo, J. Fernández, and C. Hervás-Martínez, "Can machine learning techniques help to improve the common fisheries policy?" in *Proceedings of the International Work Conference on Artificial Neural Networks* (Lecture Notes in Computer Science), I. Rojas, G. Joya, and J. Cabestany, Eds. Springer, 2013, pp. 278–286.
[41] S. Wang and X. Yao, "Multiclass imbalance problems: Analysis and potential solutions," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

María Pérez-Ortiz was born in Córdoba, Spain, in 1990. She received the B.S. degree in computer science in 2011 and the M.Sc. degree in intelligent systems in 2013 from the University of Córdoba, Córdoba, Spain, where she is currently pursuing the Ph.D. degree in computer science and artificial intelligence in the Department of Computer Science and Numerical Analysis.

Her current interests include a wide range of topics concerning machine learning and pattern recognition.

Pedro Antonio Gutiérrez (M'08) was born in Córdoba, Spain. He received the B.S. degree in computer science from the University of Sevilla, Sevilla, Spain, in 2006, and the Ph.D. degree in computer science and artificial intelligence from the University of Granada, Granada, Spain, in 2009.

He is currently an Assistant Professor with the Department of Computer Science and Numerical Analysis, the University of Córdoba, Córdoba, Spain. His current research interests include pattern recognition, neural networks, evolutionary computation, and their application to real-world problems.

César Hervás-Martínez (M'08) was born in Cuenca, Spain. He received the B.S. degree in statistics and operating research from the Universidad Complutense, Madrid, Spain, in 1978, and the Ph.D. degree in mathematics from the University of Seville, Seville, Spain, in 1986.

He is currently a Professor with the Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain, in the area of computer science and artificial intelligence and an Associate Professor with the Department of Quantitative Methods, School of Economics. His current research interests include learning algorithms, neural networks, pattern recognition, and the modeling of natural systems.

3.2. Classification of EU countries' progress towards sustainable development based on ordinal regression techniques

Since the work of [85], the interest in sustainable development (SD) has been increasingly growing in the political arena and social consciousness. Usually, sustainable development is stated to be concerned with ensuring long-term human well-being, which necessarily involves confronting the challenges of limited natural resources and global poverty, having a good standard of living, a long and healthy life, access to education, participation in the social and political life of the community and well-paid work that provides people with the opportunities to achieve their goals, hopes and aspirations [107].

A great deal of measurement attempts have been developed over the last two decades at various levels (international organisations, academic and private initiatives) for managing and monitoring progress towards SD [13, 89], some of which have focused on households, distribution of wealth, quality of life, social progress and ecological sustainability [48, 59], but without a consensus on which are the most determinants factors [70].

Because of this need, the following paper presents a preliminary study to monitor the progress of the EU toward SD and to validate the different indicators in the literature via different machine learning techniques. The aim of this work can be said to be threefold. The first aim is to perform a hierarchical clustering analysis using the 19 Eurostat official indicators of each EU country for 5 different years. The resulting clustering structure should be of help to the expert to group the country-year observations on different clusters and to rank the clusters according to their overall SD performance. The second stage of the study is based on the creation of a learning model able to measure and monitor the SD advances, by means of ordinal classification. To do so, a decomposition method is used, in conjunction with a trainable decision rule, where the order information is incorporated. Finally, the last objective is related to the interpretation of one of the ensemble models obtained in such a way that we can provide valuable information about the most relevant variables in relation to SD progress.

The clusters obtained from the first objective have been ordered according to their overall SD performance (advanced, followers, moderate and initiated). Additionally, the characterisation of these clusters made by the expert reflects a global picture of the SD stage of countries, which could enrich and complement the judgement of stakeholders more than a single indicator score value or trying to find the SD readiness of a country through separate indicators. The empirical results in this paper also indicate that the constructed model is able to achieve very promising and competitive performance. Thus, it could be used for monitoring the progress towards SD of the different EU countries, in a manner

similar to that used for rankings.

Concerning the algorithm used, decomposition methods seem to perform well for this specific application, and the results are improved by the use of the proposed trainable combiner function which makes use of the ordering of the classes.

Furthermore, the decomposition method based on logistic regression has been used again with model interpretation purposes, providing valuable information about the most relevant indicators for ranking the end-point variable. These indicators are the labour productivity per hour worked, the electricity consumption of households and the transport greenhouse emissions. In regards to the scenarios, the most important are sustainable consumption, demographic changes, global partnership and climate change and energy, a result that is in line with the three dimensions of SD.

Summarising, although it is difficult to assess the direct impact of the indicators on the progress towards sustainability, it can be stated that given the good generalisation performance of the methodology, it may be useful to monitor national strategies for European governments, in a manner similar to that used for rankings (because of the ordinal nature of the clusters obtained).

Knowledge-Based Systems 66 (2014) 178–189

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Classification of EU countries' progress towards sustainable development based on ordinal regression techniques

M. Pérez-Ortiz [a,*], M. de la Paz-Marín [b], P.A. Gutiérrez [a], C. Hervás-Martínez [a]

[a] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[b] Department of Management and Quantitative Methods, Loyola University, Córdoba, Spain

ARTICLE INFO

Article history:
Received 13 January 2014
Received in revised form 6 April 2014
Accepted 25 April 2014
Available online 9 May 2014

Keywords:
Sustainable development
European Union
Machine learning
Ordinal regression
Ensemble methods

ABSTRACT

Sustainable development (SD) is a major challenge for nations, even more so in the current economic crisis and uncertain environment. Although different indicators, compindices and rankings to measure and monitor SD advances at the macro level exist, the benefits for stakeholders and policy makers are still limited because of the absence of predictive models (in the sense of models able to classify countries according to their SD advances). To cope with this need, this paper presents a first approximation via machine learning techniques. First, we study the SD stage of the 27 European Union Member States using information from the years 2005–2010 and different major indicators that have been related to SD. A hierarchical clustering analysis is conducted, and the patterns are categorised as advanced, followers, moderate and initiated, according to their progress towards SD. The classification problem is addressed from an ordinal regression point of view because of the inherent order among the categories. To do so, a reformulation of the one-versus-all scheme for ordinal regression problems is used, making use of threshold models (Logistic Regression (LR) and Support Vector Machines in this case) and a new trainable decision rule for probability estimation fusion. The empirical results indicate that the constructed model is able to achieve very promising and competitive performance. Thus, it could be used for monitoring the progress towards SD of the different EU countries, in a manner similar to that used for rankings. Finally, the decomposition method based on LR is used for model interpretation purposes, providing valuable information about the most relevant indicators for ranking the end-point variable.

1. Introduction

Sustainable development (SD) is among the most relevant and pressing challenges of the modern age. Since the work of Meadows et al. [25], the interest in this problem has been increasingly growing in the political arena and social consciousness. The underlying idea still remains: human consumption is outstripping what the planet can produce, as we are spending natural resources faster than they can be replenished. In this sense, the academic community, main stakeholders and the political and media debate display special interest in achieving SD as a model of growth for nations and as a primary goal. In times of a deep economic crisis, society and policy-makers focus their attention on economic indicators such as income and employment rates; however, sustainability and social inclusion should also be a priority.

Although there is no consensus in what SD really is (for a discussion of sustainable development definitions see Moldan et al. [27]), a large list of definitions has been published [29,11], and this term is considered a 'contested concept' [18]. However, SD can be said to be known worldwide, thanks to the World Commission on Environment and Development, as development that 'meets the needs of the present without compromising the ability of future generations to meet their own needs' [38].

Nevertheless, what is commonly established is that SD is concerned with ensuring long-term human well-being, which necessarily involves confronting the challenges of limited natural resources and global poverty, having a good standard of living, a long and healthy life, access to education, participation in the social and political life of the community and well-paid work that provides people with the opportunities to achieve their goals, hopes and aspirations [37].

The imperious need for reliable and pertinent indicators, to better monitor and foster SD and to guide this SD process at a

* Corresponding author. Address: Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain. Tel.: +34 957 218 349; fax: +34 957 218 630.
E-mail addresses: i82perom@uco.es (M. Pérez-Ortiz), mpaz@uco.es (M. de la Paz-Marín), pagutierrez@uco.es (P.A. Gutiérrez), chervas@uco.es (C. Hervás-Martínez).

http://dx.doi.org/10.1016/j.knosys.2014.04.041
0950-7051/

national level, was recognised early at the time of the Rio Conference and the Agenda 21 [36], followed by the Commission on Sustainable Development work programme on indicators. The most common effect of indicators could be calling attention to an existing problem. However, these indicators yield different scores and rankings depending on the nature and type of assessments. They also report on past performance [3] and they do not predict whether a certain country is heading (or could head) a group in these terms.

A great deal of measurement attempts have been developed over the last two decades at various levels (international organisations, academic and private initiatives) for managing and monitoring progress towards SD [6,10,31,21,30], some of which have focused on households, distribution of wealth, quality of life, social progress and ecological sustainability [4,15,17] but without a consensus on which one is the most determinant at a general level [19].

The fact that governing bodies hold SD as a central strategy reveals the relevance of achieving this development model. In this context, governments need to incorporate SD values and principles in education systems, invest in Research and Development (R&D) to innovate in new technologies (renewal energies, new ways of ecological or organic agriculture, exploitation of scarce natural resources, air contamination reduction, waste treatment, pollution, green buildings, poverty eradication and other such goals) if they wish to progress towards SD and increase human well-being [32]. Currently, different measurement tools, models and methodologies are being used to help stakeholders to analyse the advances of countries in such direction. These tools facilitate the adoption of policies as well as the creation of national systems for a global evaluation.

SD can also be found in the core of EU priorities. According to the Foundational Treaty, the EU institutions work for the achievement of SD in Europe balancing economic growth and price stability, aiming at full employment and social progress and a high level of protection and improvement of the quality of the environment.

The current main instrument of the EU is the long-term Sustainable Development Strategy (EU SDS) [12] and a set of SD indicators developed to evaluate the progress towards SD [13]. These indicators are the main source of information used for the analysis conducted in this paper. There is much in common between well-being and SD indicators approaches and most research on both subjects has concentrated on identifying, developing, and refining criteria and indicators [33].

Among these initiatives, actions to improve and complement the current growth measurements [17] are frequently outlined. This is important, as there are various dimensions that can be interlinked, e.g. education or employment quality can affect health, social relations and status, civic participation, etc. The motivation for proposing an alternative methodology to composite indicators or indices could be that, in the first place, indices summarise too much and provide less information than the description of the characteristics of a cluster or the analysis of models able to predict the class for a new pattern. On the other hand, these indices have been found to be very sensitive to the choices of the index's construction and the selection of the variables to be used.

Because of this need, this paper presents a preliminary study to monitor the progress of the EU toward SD and to validate the different indicators in the literature via different machine learning techniques. The aim of this work can be said to be threefold. The first objective is to conduct a hierarchical clustering analysis to detect behavioural patterns in the EU country-year observations analysed. For this step, 19 Eurostat related official indicators are used, which represent major scenarios and are reported as informative for the progress towards SD in the EU SDS. From this first analysis, we obtained a hierarchical tree of the country-year observations, whose nodes are fused by an expert (examining their characteristics) obtaining four differentiated clusters. Moreover, an ordering for the different resultant clusters is also proposed according to their overall SD performance. In order to obtain a model for deciding the cluster of new countries and to measure and monitor their SD advances, the second stage of the proposed methodology applies ordinal regression to model the categories obtained from the clustering analysis. Given the ordinal nature of the classification problem, the next objective of the paper is to assess the learning problem from an ordinal regression perspective (to benefit from this order information). To do so, we reformulate the decision rule of a recently proposed ordinal ensemble algorithm [28] to obtain more robust results and accurately predict the progress towards SD. The last objective is related to the interpretation of one of the ensemble models obtained in such a way that we can provide valuable information about the most relevant variables in relation to SD progress.

Although the application of a classification algorithm after a clustering process could seem to be trivial, there are three reasons why, in our case, the derived models contribute interesting information: (1) the fact that we are considering expert knowledge to fuse some of the clusters derived from the hierarchical clustering, (2) the ordering given to the resulting groups together with the application of ordinal regression, which provide experts with a ranking tool able to classify new evaluated countries in the groups discovered and to rank them (even when they belong to the same group) and (3) the extraction of information about the most important indicators to characterise and differentiate sustainable developers countries from non-sustainable developers countries (rather than just validating the performance of machine learning algorithms). Note that, for this interpretability analysis, the clustering algorithm itself is not usually very helpful.

Ordinal regression (also known as ordinal classification) problems arise in fields as information retrieval, preference learning, economy, and sociology and nowadays it is considered as an emerging field in the areas of machine learning and pattern recognition. Ordinal regression could be said to be a relatively new learning paradigm which shares properties of classification and regression. Formally, for this problem, $\mathcal{Y}$ (the labelling space) is a finite set, but there exists some ordering among its elements. In contrast to regression, $\mathcal{Y}$ is a non-metric space, thus distances among categories are unknown. Besides, the standard zero-one loss function does not reflect the ordering of $\mathcal{Y}$.

A great number of statistical methods for categorical data treat all response variables as nominal, in such a way that the results are invariant to order permutations on those variables. However, there are many advantages in treating an ordered categorical variable as ordinal rather than nominal [1,22], a statement applicable to classification problems. In this vein, several approaches to tackle ordinal regression have been proposed in the domain of machine learning over the years, since the first methodology (the Proportional Odds Model or POM) dating back to 1980 [24]. Indeed, the most popular approach in this paradigm is the use of threshold models, which are based on the assumption that an underlying real-valued outcomes exist although they are unobservable.

For the learning process, a recently proposed ordinal ensemble methodology [28] based on threshold models is used, and a new trainable decision rule is developed to simplify and make it more robust than the initial proposal. The ensemble methodology is based on decomposing ordinal regression problems into simpler classification tasks, where the order information is explicitly included. The main hypothesis is that the performance of any ordinal algorithm could be improved by simplifying the original classification problem and formulating multiple order hypotheses (hypotheses that will be combined in a final decision function). Therefore, the decision function choice is a crucial step. However,

the original proposal makes use of a fixed decision function that is non-trainable and does not take the data labelling into account [28]. In contrast, in this paper we develop a class conscious or trainable approach (i.e., we learn a weight vector for each member of the ensemble) to obtain more robust results.

After this introductory Section 1, the methodology applied in this study is set out in detail in Section 2; in Section 3, the dataset and the experimental setup is described and the main results are discussed. Finally, the conclusions are drawn in Section 4 along with possible future lines of research.

2. Methodology

The methodology used in this paper presents two differentiated parts. First, a clustering analysis is performed to discover the main groups of country-year observations. The resultant clusters are studied and ranked from the point of view of sustainable development. In the second phase, an ordinal classifier is considered for the learning phase and for its interpretation.

2.1. Hierarchical clustering methodology

As previously mentioned, the first step of this study corresponds to a clustering analysis to identify the EU country-year cases that can be said to be similar with respect to their overall SD performance. This clustering is based on nineteen related indicators, which are grouped by scenarios. In the absence of supervised class labels or knowledge about the clusters, as in the case presented, it is difficult to select a criterion to judge whether one clustering algorithm is performing better than another. Because of this, we make use of a hierarchical clustering technique (more specifically the Cobweb algorithm), which provides insight into the data by assembling the objects into a dendogram and creating a hierarchy that could be more informative than an unstructured set of clusters.

Different SD indicators have been analysed and grouped before in the literature via a hierarchical clustering algorithm to identify regions or patterns with similar behaviour [23,2] with the main aim of ranking the countries and developing composite or aggregate indicators. However, these studies only consider one year and they do not describe the characteristics associated to each cluster as in the present work.

2.2. Ordinal regression methodology

In this subsection, a new ordinal regression algorithm based on a class conscious decomposition of the label space is presented for the learning process. This method will be used for the second part of the study.

The goal in classification is to assign an input vector $\mathbf{x}$ to one of K discrete classes $\mathcal{C}_k, k \in \{1,\dots,K\}$. A formal framework for the ordinal regression problem can be introduced considering an input space $\mathcal{X} \in \mathbb{R}^d$, where d is the data dimensionality. An outcome space $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ is defined, where the labels are ordered (i.e. $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \dots \prec \mathcal{C}_K$, where $\prec$ denotes this order information). The objective then is to find a prediction rule $f : \mathcal{X} \to \mathcal{Y}$ by using an i.i.d. training sample $T = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ where N is the number of training patterns. For convenience, denote by $\mathbf{X}_i$ to the set of patterns belonging to $\mathcal{C}_i$.

Threshold methods are based on the idea of modelling ordinal regression problems from a regression perspective, by assuming an underlying real-valued outcome (also known as latent variable), which is unobservable. Consequently, these methodologies try to estimate:

- A function $f(\mathbf{x})$ to predict the nature of the underlying real-valued outcome.
- A set of thresholds $\mathbf{b} = \{b_1, b_2, \dots, b_{K-1}\} \in \mathbb{R}^{K-1}$ to represent the intervals in the range of $f(\mathbf{x})$, where $b_1 \leqslant b_2 \leqslant \dots \leqslant b_{K-1}$.

Two of these threshold methods (selected for the ensemble) are now briefly described.

The SVM paradigm [8] is considered the most common kernel learning method for statistical pattern recognition. Some works in the SVM literature have been focused on the reformulation of this successful paradigm to tackle ordinal regression problems [16,34,7]. All these approaches share one common objective which is the definition of $K-1$ discriminant hyperplanes represented by the vector $\mathbf{w}$ and the scalars bias $b_1 \leqslant \dots \leqslant b_{K-1}$ in order to properly separate training data into ordered classes by modelling ranks as intervals on the real line.

In machine learning and pattern recognition, LR [26] is a well-known methodology based on a regression analysis for classification tasks. This method has been reformulated to deal with ordinal regression tasks giving rise to the so-called Proportional Odds Model (POM) [24]. This model was the first threshold method applied to ordinal classification problems and it is based on a linear projection which is jointly trained with a set of thresholds by using a similar strategy to that considered for nominal LR.

2.2.1. Ensemble for ordinal regression

Nowadays, the ensemble paradigm, which tries to imitate human nature to seek several opinions before making a crucial decision, is one of the most actively researched in pattern recognition and machine learning [20]. It is usually considered as an alternative to the conventional "standalone" methods, which may be suboptimal.

Concerning ordinal regression, a great deal of the methodologies proposed can be categorised under the name of decomposition methods (which could be also seen as ensemble methods in the sense that they decompose the original ordinal regression learning problem into several binary classification tasks); e.g. creating a new outcome variable that corresponds to the question: "is the label of pattern $\mathbf{x}$ greater than k?", for some given rank k.

In this sense, it has been recently demonstrated that the decomposition of the target variable into several binary and ordinal classification problems can improve the results for different ordinal classification tasks [28]. That methodology can be seen as a reformulation of the one-versus-all scheme for ordinal labels. Let us formally define the approach: Given K different ordered classes $(\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K)$, the idea is to compute K different classification problems by performing a supervised relabelling of the data (and training the learning algorithm with these relabelled patterns). In particular, we will obtain K different decision makers $\mathbb{D} = \{D_1, \dots, D_K\}$, where D_i will focus on separating the corresponding class $\mathcal{C}_i$ from previous classes $(\mathcal{C}_1, \dots, \mathcal{C}_{i-1})$ and subsequent classes $(\mathcal{C}_{i+1}, \dots, \mathcal{C}_K)$. Therefore, the set of events to classify for decision maker D_i can be defined as:

$$\{\mathcal{G}_{i1} = (\mathcal{C}_1 \cup \dots \cup \mathcal{C}_{i-1}), \mathcal{G}_{i2} = \mathcal{C}_i, \mathcal{G}_{i3} = (\mathcal{C}_{i+1} \cup \dots \cup \mathcal{C}_K)\}. \quad (1)$$

Note that for decision makers D_1 and D_K, the events $\mathcal{G}_{i1}$ and $\mathcal{G}_{i3}$ are respectively the empty set (thus they compute standard binary classification tasks). However, for the case of $\{D_2, \dots, D_{K-1}\}$ an ordinal algorithm should be used due to the natural ordering of the three events.

Since this method assumes the application of threshold methods, each decision maker D_i will be composed of a projection $\mathbf{w}_i$ and a set of thresholds $\mathbf{b}_i$ separating the associated projected events. This corresponds to step one in Fig. 1, where the whole process is summarised.

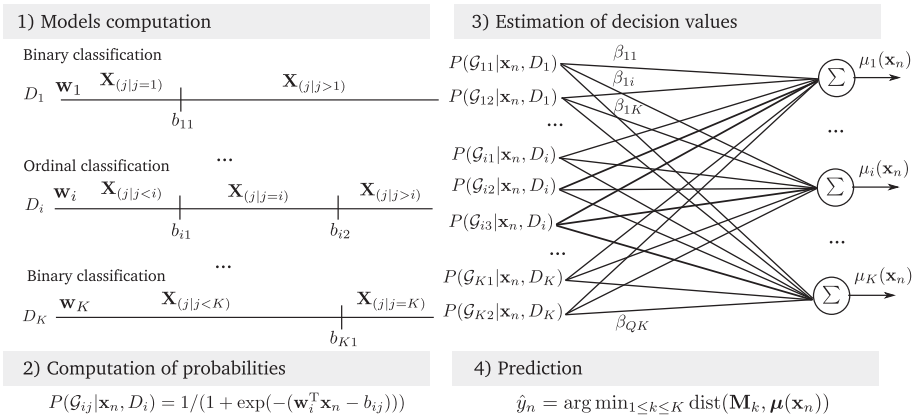


Fig. 1. Different steps for the ordinal ensemble proposed.

The choice of threshold models as base classifiers for the ensemble is not arbitrary. This choice is justified because of their common use in ordinal regression problems and because of their inherent advantage to lend themselves to probabilistic outputs (and it is clear that these conditional probabilities of class membership will be useful for constructing a more robust ensemble). For each pattern and decision maker D_i, the probability of belonging to class $\mathcal{C}_i$ (or event $\mathcal{G}_{i2}$) will be calculated, along with the probability of belonging to the previous event $\mathcal{G}_{i1}$ and the probability of belonging to the following event $\mathcal{G}_{i3}$. This corresponds to step two in Fig. 1. Note that the logit function has been chosen for the probability estimations:

$$P(\mathcal{G}_{ij}|\mathbf{x}_n, D_i) = \frac{1}{1+\exp\left(-\left(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_n - b_{ij}\right)\right)}. \tag{2}$$

2.2.2. Proposed class conscious trainable ensemble

The next step of the algorithm is to choose a decision function that joins all the probabilistic information obtained up to this point. In the original proposal [28], the probability for a specific event $\mathcal{G}_{ij}$ is distributed equally among all the classes involved, and a probability combination function is then used (such as the sum or the product). Furthermore, a weighting procedure is also needed because this distribution results in some classes receiving more attention and these will then be more probable than the rest. The choice of how to distribute the probabilities among the classes and the weights to apply is a complex issue if a non-trainable method (like the one in the original proposal) is applied. Because of this, we propose the use of a so-called class conscious or trainable ensemble method. This strategy will optimise a weight vector according to the original target values.

After training K decisors for the different events in Eq. (1), we will have a total of $3K-2$ probability estimations because two of the decisors are binary and all the others are 3-class classifiers, see Fig. 1. These probabilities can be expressed as $P(\mathcal{G}_{ij}|T, D_i)$, $i = \{1,\ldots,K\}$, where $j = \{1,2,3\}$ for 3-class classifiers, $j = \{1,2\}$ for binary ones and T represents the whole training set. Now, consider $\mathbf{H} \in \mathbb{R}^{N\times Q}$ as the matrix storing the estimated probabilities for the different events and decisors as columns, and the different patterns as rows. In this sense, $\mathbf{H}$ contains the probability for each pattern $\mathbf{x}_n \in T$, $n = \{1,\ldots,N\}$ of belonging to the concrete event j estimated by the decisor i, i.e., $\mathcal{G}_{ij}$ (the total number of combinations being $Q = 3K-2$). For convenience, for a specific event q and pattern $\mathbf{x}_n$ this probability will be represented by h_{nq}, $q = \{1,\ldots,Q\}$. Our objective is to combine all these probabilities in the best possible way to improve the final ensemble performance, which can be accomplished by considering a weighting procedure. Consequently, we propose to calculate the support for class $\mathcal{C}_i$ as follows by using a weighting matrix $\boldsymbol{\beta} \in \mathbb{R}^{Q\times K}$:

$$\mu_i(\mathbf{x}_n) = \sum_{q=1}^{Q} \beta_{qi} \cdot h_{nq}. \tag{3}$$

To optimise $\boldsymbol{\beta}$ we first need to define a coding matrix $\mathbf{M} \in \mathbb{R}^{K\times K}$ representing the output labelling information, where for simplicity $\mathbf{M}_k$ will denote the code associated to class $\mathcal{C}_k$ (i.e., k-th column of the coding matrix). For this, we considered two approaches: the nominal one-versus-all matrix $\mathbf{M}^{(n)}$ and an ordinal coding matrix $\mathbf{M}^{(o)}$. These matrices can be seen for a 4-class problem in Table 1, where rows correspond to the binary subproblems and columns to the role of each class for each subproblem.[1] The label $+1$ is associated to the positive class and -1 to the negative one. At this point, denote by $\mathbf{A} \in \mathbb{R}^{N\times K}$ as the matrix storing the real coded output for each pattern in the training set (i.e., for a given pattern $\mathbf{x}_n \in \mathbf{X}_k$ the associated code will be $\mathbf{M}_k$). This matrix $\mathbf{A}$ will be used for optimising $\boldsymbol{\beta}$ and constructing our class-conscious ensemble.

The system to solve is: $\mathbf{H}\cdot\boldsymbol{\beta} = \mathbf{A}$, where the weights are commonly derived using linear regression, i.e.:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{A}, \tag{4}$$

† denoting the Moore–Penrose pseudoinverse.

For the prediction phase, we compute the Euclidean distance of the decision values $\boldsymbol{\mu}(\mathbf{x})$ for pattern $\mathbf{x}$ to each coding $\mathbf{M}_k$, $k = \{1,\ldots,K\}$. The predicted class is then chosen so as to provide the minimum distance:

$$\hat{y}_n = \arg\min_{1\leq k\leq K} \mathrm{dist}(\mathbf{M}_k, \boldsymbol{\mu}(\mathbf{x}_n)), \tag{5}$$

where $\hat{y}_n$ is the label predicted for $\mathbf{x}_n$. For a graphical summary of the proposal see Fig. 1.

3. Experimental study and results

This section presents the experimental study conducted in this paper for the problem considered and analyses the results obtained. We first describe the variables chosen and then the clustering process and groups derived. Finally, two different experiments are conducted to analyse the goodness of the proposed classification technique.

[1] Note that for the ordinal coding matrix the first row of $\mathbf{M}$ is unnecessary. However, we assume it for the sake of homogeneity in the notation for the nominal and ordinal cases.

Table 1
Example of coding matrices considered for a 4-class ordinal problem.

Nominal approach $\mathbf{M}^{(n)}$	Ordinal approach $\mathbf{M}^{(o)}$
$\begin{pmatrix} +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 \\ -1 & -1 & +1 & -1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$	$\begin{pmatrix} +1 & +1 & +1 & +1 \\ -1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & -1 & +1 \end{pmatrix}$

3.1. Variables description

The selection of proper indicators for the problem considered can be said to be a difficult issue, especially when there is no common understanding of well-being or sustainability. Consequently, despite the myriad of SD indicators available, the official set of indicators employed by European Council to monitor the progress in the EU SDS was selected for this research study. According to the last report of the EU SDS, there are more than 100 SD indicators, 11 of which 'have been identified as headline indicators' to offer an overall picture of the progress towards SD in terms of the targets defined in the EU SDS [13]. The selected indicators are presented in Table A.8 by scenario. The data has been collected from Eurostat official website[2] considering the 27 EU Member States. Although these variables can show certain degree of correlation, the capacity control of SVM classifiers has been found to be equivalent to some form of regularisation [35] and therefore alleviates this problem.

According to the last SD Report [12], the variables selected can be classified in various scenarios (omitting the natural resources scenario because of the lack of data for common bird index values, status of fish stocks and good governance scenario because of their quality nature). These indicators have been grouped into four domains: social, economic, environmental and institutional-political. Within each domain, the different topics have been separated in scenarios constituting the reflection of SD aims and priorities.

Because of data availability (until 2010 in Eurostat official data base), 162 patterns were selected considering each country and year for the period 2005–2010 as a single composed item (country-year observation).

3.2. Clustering results

The Cobweb hierarchical clustering algorithm was applied to the dataset described in the previous subsection. The resulting dendogram can be seen in Fig. 2. For each node, the identifier of the cluster corresponds to the number without brackets and the number in brackets is the number of country-year patterns included in the corresponding cluster. Some of the clusters in Fig. 2 are the result of a merging procedure of different nodes of the original hierarchical tree. This merging was done according to the knowledge of the expert (subjective and objective knowledge of the current economic, sociological and environmental situation of the different European members). Note that other clustering algorithms were tested, such as the well-known *k*-means method, which was discarded given the inconsistent nature of the clusters obtained (from the point of view of the progress towards SD).

Now, we present a description, justification and characterisation of the clusters obtained from the previous analysis along with an ordering among them. The description is based on the most relevant variables and scenarios. Note that as the analysis has been performed employing country-year observations. Therefore, if the variables values change in the time span considered, the country-year patterns could also change or move to a more suitable cluster.

[2] http://epp.eurostat.ec.europa.eu/portal/page/portal/sdi/indicators.

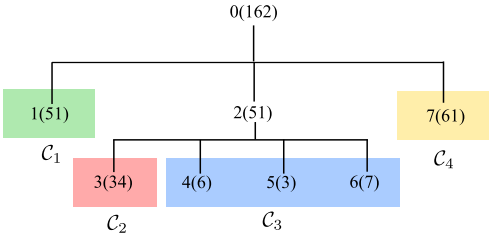


Fig. 2. Dendogram obtained from hierarchical clustering.

This is the reason why some countries are included in more than one cluster, but for different years. The procedure followed to order the clusters is described in detail in the following subsection.

3.2.1. First cluster: advanced ($\mathcal{C}_1$)

The country-year patterns involved in this cluster are the following: Austria, Denmark, Finland, France-05-07-08-09-10, Germany, Ireland-05-06-07-08, Netherlands, Sweden and United Kingdom. This cluster groups very prosperous, democratic countries, with well-developed and modern market economies, skilled labour forces, and a high standard of living with very high per capita output. They are also characterised by extensive government welfare measures, social security systems (currently challenged by low fertility growth and rapidly ageing population), an equitable distribution of incomes, historically low levels of unemployment and they are also well-known because of the high standard of their education systems.

They present healthy budget surplus for many years up to 2008, but the recession affected general government finances and the debt ratio. Thus, the budget balance swung into deficit in 2009 (stabilisation measures, stimulus spending and an income tax reform pushed the budget deficit). Their fiscal position compares favourably with other euro-zone countries, but it faces external risks such as political and economic uncertainties caused by the European sovereign debt crisis.

Although they present a low economic growth rate due to their high levels of GDP, they display a high level of productivity per hour of work growth (a sign of competitiveness), the highest levels of employment in the EU and they are by far the countries that devote the greatest amount of funds to Research and Development (R&D).

These countries are very efficient in the use of their resources, represented by the share of Domestic Material Consumption (DMC) by unit of GDP. Regarding the area of sustainable consumption and production, they are leaders in the share of total utilised agricultural area occupied by organic farming and in the quantity of electricity consumed by households, which depends on economic activity, national customs and even national weather.

Concerning the scenario of demographic changes, the employment rate of older workers is also the highest one. As said before, the government finances and the debt ratio are affected by the recession, turning previously strong budget surpluses into deficits. For this reason, this group is the second one in this indicator, although far from the first one, namely, the second cluster.

Most likely due to their levels of prosperity and R&D investment, these are the countries with the highest share of renewable energy in gross final energy consumption. Moreover, they have the second position in lowest greenhouse gas (GHG) emissions and they perform the best as they present the lowest ratio between transport energy consumption and GDP. Thus, they do not pollute as much as might be thought because of their level of GDP and prosperous economic activity on a first analysis. Nevertheless, it is the group that has the highest level of GHG emissions from transport as regulated in the Kyoto Protocol.

Finally, in the scenario of global partnership, they are the first in official development assistance as share of gross national income and EU Imports from developing countries indicators, which is logical, as these countries present the highest levels of GDP.

3.2.2. Second cluster: followers (C_2)

The country-year patterns involved in this cluster are the following: Belgium, Czech Republic-10, France-06, Greece, Ireland-09-10, Italy, Luxembourg and Spain. This cluster mainly groups traditionally relevant economies (Mediterranean countries and the two EU Member States of the BENELUX, i.e., Belgium and Luxembourg, along with some country-year patterns), although not as relevant as the ones in the previous cluster. Most of these countries have been strongly shocked by the international economic crisis. For example, in the three Mediterranean countries, tourism provides an important share of GDP (which has been severely affected by the consumption fall) and their recovery appears to be experiencing more difficulties than the previous cluster.

This is the cluster with the lowest average economic growth, but it is ranked second in labour productivity per hour worked, a consequence of the decrease in the employment rate (this cluster rank the third in this indicator: GDP grows with less labour force).

They tried to maintain social equity by means of laws, tax policies and social spending, reducing income disparity and the impact of free markets on public health and welfare, but public debt is very high (they rank the first, far from the following clusters) and the governments have approved the most severe austerity packages consisting of expenditure cuts and new revenues.

In the sustainable production and consumption scenario, these countries are the most productive in the use of their resources along with the first cluster (almost equal average of resource productivity indicator). We find it in third position in area under organic farming but very close to the two previous clusters in the order. According to their levels of GDP and standards of living, these countries are ranked second in electricity consumption but are 1.5-fold lower than the first cluster.

Nevertheless, in terms of the climate change scenario, their performance is low. These are the countries with the highest level of GHG emissions (especially Spain, Greece and Italy), and their share of renewable energy is low (except Spain and Greece) and very similar to the fourth cluster. They are ranked third in transport energy consumption, although this indicator presents a low standard deviation, and second in GHG emissions from transport, very close to the first cluster.

3.2.3. Third cluster: moderate (C_3)

The country-year patterns involved in this cluster are the following: Bulgaria-05-06-07, Hungary-07, Malta and Poland. This cluster groups three former communist countries along with Malta (in the case of Bulgaria, 2005, 2006 and 2007 years and in the case of Hungary, only the 2007 year). This cluster groups countries with the highest economic growth in terms of GDP (mean = 3.81%). As new economies, they present this growth due to the abundance of natural resources and a high domestic demand following a period of socialism, among other reasons. Nonetheless, their economies do not seem to be based either on education or on R&D investment. Poland is a special case because its transition to capitalism was facilitated by the fact that its government tolerated a small number of private firms. The economic profile of this group has the peculiarity that its economic performance is the worst in the rest of indicators (i.e., labour productivity per hour worked and a very low amount of resources devoted to R&D as a percentage of GDP). They also present almost the same high value, on average, of gross public debt as cluster 2.

They present a low performance in the sustainable consumption and production scenario. Their third position in the resource productivity indicator, in the electricity consumption of households and their last position in the area under organic farming indicator, far from its mean value, pinpoint the reasons for low performance.

Intermediate performance also appears if we look at the climate change scenario: it is the third cluster in GHG gas emissions, mainly because of their economic growth and their low share of renewable energy in gross final energy consumption. The emission increases are at least partly due to colder winter months as well as emissions from electricity and heat production.

In the global partnership scenario, they are the last in the official development assistance as share of gross national income and EU Imports from developing countries indicators, as these are the countries with lowest GDPs along with the next fourth cluster. The values are far from clusters 1 and 2. As a final remark, from the hierarchy obtained from the clustering analysis, it can be seen that clusters 2 and 3 are more similar to each other than to the other clusters because they arise from the same tree node, and this fact ensures even more the ordering between these classes.

3.2.4. Fourth cluster: initiated (C_4)

The country-year patterns involved in this cluster are the following: Bulgaria-08-09-10, Cyprus, Czech Republic-05-06-07-08-09, Estonia, Hungary-05-06-08-09-10, Latvia, Lithuania, Portugal, Romania, Slovakia, Slovenia. The majority of these countries (as well as those in the previous cluster) were post-socialist economies. They present the peculiarity that they have a population of approximately ten million people or less. Thus, they have in common high economic growth, but lower than the previous group, and the lowest GDP per capita of the EU. Their economic performances are better in the rest of indicators (e.g., the third in labour productivity per hour worked). Moreover, they perform better in the demographic changes scenario than the previous cluster.

Most likely due to their more reduced social aids and public support, they present the lowest level of gross public debt. They display a medium–low performance in the sustainable consumption and production scenario, and they are in the last position in the resource productivity indicator and in the electricity consumption of households, far from the values of the previous clusters (low GDP per capita and low population may be the reasons for this level of electricity consumption of households). The low electricity consumption (sevenfold difference from the first cluster) could also reflect differences in lifestyles, habits, climate and the lower use of electronic devices because of the low GDP per capita, among other reasons. They differ from the previous cluster in the fact that they are the second group in relation to the area under organic farming. A good performance can be found in the climate change scenario: this cluster has fewer GHG emissions and is the second cluster, in addition to being very close to the first cluster, in the share of renewable energy in gross final energy consumption.

At this point, it must be said that although the objective values reveal a good performance, they must be reinterpreted because low industrial activity and low final energy consumption could lead to the wrong conclusions. Final energy consumption is essential for development, but its use may result in greenhouse gas emissions. If this cluster presents a low GDP per capita and social development, this could be the reason for low greenhouse gas emissions more than a real improvement in energy efficiency.

3.3. Order of the clusters

The ordering for the different clusters (i.e. C_1 or advanced, C_2 or followers, etc.) was derived using the following steps:

1. Analysis of the different indicators (some of them can be considered as more relevant than others when referring to sustainable development, e.g. the key indicators).
2. Analysis of each of the resultant clusters, studying the country-year observations and the statistics values for the different indicators (mean, variance, maximum and minimum).
3. Order the clusters according to the information obtained in the previous step, complemented with different sustainable development and environmental reports and country profiles (as the ones provided by the European Environmental Agency) and useful information for some socioeconomic and historical aspects of the selected countries (mainly from the CIA world factbook). Table 2 shows the order of the mean value for the clusters considering the nineteen selected indicators (in descendent order), which was one of the sources of information used for ranking the classes, where for example the order of clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ can be clearly appreciated. The process consists of considering all the possible ways to merge the clusters returned by Cobweb and selecting the one with the clearest ordering, given that we want to obtain a ranking tool for the countries evaluated. The results on the table are taken from the final clusters used in the paper. Note that, although the purpose is not to provide a strict monotone relation between the clusters and the different indicators, this information is useful to analyse which clusters present in general a good performance towards sustainable development, notwithstanding that some indicators may interfere with the rest and that all of them should be considered as a whole (e.g. usually non-developed countries will present a higher growth of gross domestic product per capita than developed countries). However, the order for $\mathcal{C}_3$ and $\mathcal{C}_4$ is not very clear from the table, then, other sources of information were used together with the subjective knowledge of the expert.

The exact results of the clustering can be reproduced for other studies considering the country-year information specified in each cluster description (when the year is not included in the description, all the considered years are grouped in this cluster).

Table 2
Class order for the mean of the different indicators used.

Indicator	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$
Scenario 1: socioeconomic development				
GDPGR	3	4	1	2
RLPGH	1	2	4	3
EMPLO	1	3	4	2
RDEXPE	1	2	4	3
Scenario 2: sustainable consumption and production				
RESPROD	1	2	3	4
ORGAN	1	3	4	2
ELECT	1	2	3	4
Scenario 3: social inclusion				
RISKPOV	1	2	4	3
EARLY	1	3	4	2
Scenario 4: demographic changes				
EMPOLD	1	3	4	2
PUBDE	3	4	2	1
Scenario 5: public health				
LIFEE	2	1	3	4
HELIF	3	2	1	4
Scenario 6: climate change and energy				
GREGA	2	4	3	1
RENWA	1	3	4	2
Scenario 7: sustainable transport				
TRANS	1	2	3	4
GGTRA	1	2	3	4
Scenario 8: global partnership				
ASSIS	1	2	4	3
EUIMP	1	2	3	4

3.4. Classification results

The clustering step was done only once considering the whole dataset. Then, two different experiments were undertaken. This second experimental part compares different classifiers for the task of predicting the class label derived in the previous subsection. The results of different methods were compared to evaluate which of them performed the best and to determine the ability of the ordinal ensemble methods proposed. Therefore, regarding the experimental setup, two differentiated experiments have been designed with different purposes. First, a holdout stratified technique has been applied 30 times, using 75% of the patterns for training and the remaining 25% for testing. This experiment is made to fairly compare the performance of the different classifiers considered in this study, with the aim of detecting overfitting. However, it would be unrealistic to randomly select one of the models to interpret it, because the time component of the data would be ignored. Consequently, in a second experiment, we consider three different data partitions with different years for training and testing to analyse in a more realistic situation the predictive capability of the algorithm. Moreover, the analysis of one of the models based on LR provides us with valuable information to identify the most relevant variables related to SD.

Several measures can be considered for evaluating ordinal classifiers. The most common ones in machine learning are the mean absolute error (*MAE*) and the mean zero-one error (*MZE*) [14], being $MZE = 1 - Acc$, where *Acc* is the accuracy or correct classification rate. However, these measures may not be the best option when the costs of different errors vary markedly (as in ordinal classification problems) or when the dataset is unbalanced (as in this case, see pattern distribution in Fig. 2). Because of that, this work makes use of other measure to evaluate an ordinal classifier performance. In this case, we use the maximum mean absolute error (*MMAE*), which is the *MAE* value considering only the patterns from the class with the greatest distance between true labels and predicted ones

$$MMAE = \max\{MAE_k; k \in \{1, \ldots, K\}\}, \tag{6}$$

where MAE_k is the *MAE* value considering only the patterns from the k-th class, *MAE* being the average deviation in absolute value of the predicted class from the true class [5], i.e.,

$$MAE_k = \frac{1}{N_k}\sum_{i=1}^{N_k}|y_i - \hat{y}_i|, \tag{7}$$

and N_k is the number of pattern in this class. *MMAE* values range from 0 to $K-1$. This measure was recently proposed [9] and its advantage is that a low *MMAE* represents a low error for all independently considered classes. We considered this measure due to its specific nature for ordinal and imbalanced classification problems (as the one treated in this paper).

3.4.1. First experiment: classifier comparison

For an extensive analysis, several methods are compared in this subsection. The proposed methodologies are applied using both SVM and LR (and their reformulation to ordinal regression) as base methods. From now on, the methodologies tested will be named as:

- Support Vector classifier for Ordinal Regression with Implicit Constraints (SVORIM) and its linear version (LSVORIM): Reformulation of the SVM paradigm for ordinal regression problems.

- Proportional Odds Model (POM): Reformulation of the LR paradigm for ordinal regression problems.
- Non-trainable Ensemble (NE (SVM) and NE (LR)): The original proposal [28] using both SVM and LR.
- Trainable Ensemble with Nominal Coding (TENC (SVM) and TENC (LR)) and Ordinal Coding (TEOC (SVM) and TEOC (LR)): In this case the new decision function proposed in this paper is included in the ensemble and we compare the nominal approach and the ordinal one.

The parameters of each algorithm are chosen using a nested validation with each of the training sets (k-fold method with $k = 5$). The kernel selected for all the algorithms is the Gaussian one, $K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)$ where σ is the standard deviation. For every tested kernel method, the kernel width was selected within these values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, as well as the cost parameter associated with SVM methods.

The results obtained for this first experiment are shown in Table 3, where several conclusions can be drawn. First, one can appreciate the high accuracy obtained for the dataset for most of the methodologies, which indicates that the selected variables were appropriate for the learning problem. More specifically, the ensemble methods tested display competitive results when compared to other state-of-the-art approaches. Furthermore, it can be seen that in all the cases the ensemble methods present a greater performance than the associated base methodologies. Note that although LR-based approaches make use of a linear projection (unlike the kernel methods used) they also provide reasonable results for the dataset, with the advantage that they can be used for model interpretability purposes (which will be the main purpose of the following experiment). Finally, concerning the proposed probability fusion function (comparing TEN versus NE methods), the ensemble appears to adjust better to the original labels and therefore displays a better performance. Moreover, the ordinal coding matrix (TEOC versus TENC methods) provides better results for this specific dataset (especially when we consider a metric with an ordinal nature as the *MMAE*), demonstrating this the implicit ordering present in the labelling space for the constructed dataset. It was checked that the results in Table 3 (test results) were generally very similar to the training results, so overfitting was not a problem despite the high dimensionality of the dataset.

From the results obtained for LSVORIM and POM (shown in Table 3), it can be seen that there exists a significant nonlinear component in the dataset (e.g. compare the results of these two methods with the nonlinear SVORIM technique). However, the differences when considering the ensembles with SVM and LR (a nonlinear method versus a linear one) are not very high (for example, compare the results obtained from TEOC (LR) with the results of TEOC (SVM)). To better analyse this, we have included an additional experiment with an ordinal highly nonlinear and complex synthetic two-dimensional dataset, which representation can be seen in Fig. 3.

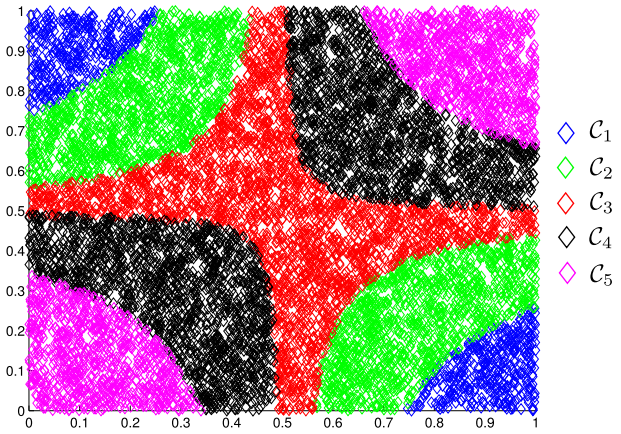


Fig. 3. Representation of the synthetic nonlinear toy dataset.

For this experiment, the same experimental setup is considered (30 stratified random partitions and cross-validation step). The results obtained in this case can be seen in Table 4 where two linear algorithms are compared with their ensemble versions and with the nonlinear version of the SVORIM method. From these results, it can be seen that the application of an ensemble technique that simplify the original classification problem is useful when dealing with highly nonlinear datasets and linear base methods (note that the ensemble methods improve the performance of the linear methods, which can be said to perform trivially given their poor performance for *Acc* and *MMAE*).

For the sake of understanding, Fig. 4 has been included, where the first step of the methodology is shown (models computation phase) for the toy dataset. Each colour represents a different class, and one model is computed to differentiate each class from the rest taking the ordering of the labelling space into account. This result is afterwards used for determining the probability of belonging to each class, and finally, we estimate the decision values and proceed to the prediction phase.

3.4.2. Second experiment: ensemble model interpretation

In this case, we use the trainable ensemble based on LR with the ordinal coding (TEOC (LR)) for the experiment. We consider three data partitions using different years for training and testing. The

Table 3
Mean ± Standard deviation results obtained (*Acc* and *MMAE* for the test sets) for the different methodologies considered.

Methodology	*Acc*	*MMAE*
State-of-the-art methods		
LSVORIM	79.593 ± 4.684	0.4618 ± 0.1349
SVORIM	91.463 ± 3.374	0.2249 ± 0.0816
POM	77.398 ± 5.727	0.4515 ± 0.1387
NE (SVM)	91.870 ± 3.966	0.2416 ± 0.1240
NE (LR)	84.065 ± 6.201	0.2922 ± 0.0880
Modified ensemble methodologies		
TENC (SVM)	**95.772 ± 3.562**	0.1379 ± 0.0991
TEOC (SVM)	**95.772 ± 3.997**	**0.1363 ± 0.0998**
TENC (LR)	87.724 ± 4.771	0.2651 ± 0.0954
TEOC (LR)	88.374 ± 4.822	0.2465 ± 0.0849

The best result is in **bold** face and the second best result is in *italics*.

Table 4
Mean ± Standard deviation results obtained (*Acc* and *MMAE* for the test sets) for the different methodologies considered and the synthetic toy dataset.

Methodology	*Acc*	*MMAE*
Linear methods		
LSVORIM	26.489 ± 0.461	2.000 ± 0.000
POM	28.933 ± 2.553	2.090 ± 0.229
Ensemble versions using linear base methods		
TENC (SVM)	60.222 ± 7.461	0.751 ± 0.260
TEOC (SVM)	60.222 ± 7.461	0.751 ± 0.260
TENC (LR)	58.178 ± 8.200	0.757 ± 0.250
TEOC (LR)	64.356 ± 7.922	0.611 ± 0.159
Nonlinear method		
SVORIM	**96.622 ± 2.034**	**0.093 ± 0.051**

The best result is in **bold** face and the second best result is in *italics*.

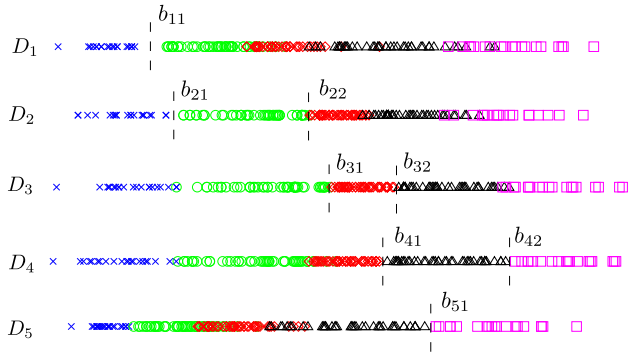


Fig. 4. Projected training results of the different base models for the ensemble (synthetic toy dataset).

first data partition (2005–2007 for training and 2008 for testing) attempts to analyse the impact of the recession transition (which fully takes place around the year 2008) on the classification model. The second data partition (2008–2009 for training and 2010 for testing) attempts to analyse the opposite effect in the model (i.e., how the model respond to training with recession years and generalising also with a recession set). Finally, we design a third experiment (training with 2005–2009 and testing with 2010) to obtain a model for the purpose of analysing the most determinant variables for the end-point variable. The results obtained for these experiments can be seen in Table 5. From these results, it can be said that, as it is obvious, the impact of the recession in the model is high, as different countries may change their associated class.

As said before, one of the methodologies chosen as base algorithm in the previous subsection (LR) can be considered as interpretable (unlike the kernel methods used for other experiments), in the sense that it can give us clues about the importance of each attribute for modelling the dependent variable and rank the different classes (it is a linear predictor). Furthermore, concerning interpretability, the decomposition proposed provides us with additional information in the sense that one model for differentiating each class from the previous and following classes is computed. Therefore, instead of being provided with a model for tackling the whole learning problem, we obtain a model for ranking each class with respect to the rest and we could analyse independently the most determinant variables. this step, we are analysing the models to rank these clusters.

The model obtained from the third data partition (i.e., using 2005–2009 for training and 2010 for testing) can be seen in Table 6. Note that both D_1 and D_4 are binary classifiers with a single threshold. The most important variables for modelling the labelling are the ones with higher $|\mathbf{w}_i|$ value. One should note that although the sign of $\mathbf{w}_i$ could also be used for an interpretability analysis, it is actually dependent on the variable coding. Furthermore, it can be seen that variables important for some of the models are not so determining for others.

To analyse the models, we should consider both the coefficients and bias obtained and the variables codification and possible values. One should take into account that, as this problem is addressed from an ordinal regression point of view, the classes will be always projected in an ordered fashion.

Generally, analysing the variable rankings in Table 7, the most influential variables for the problem are the following:

- Labour productivity per hour worked (RLPGH): This variable presents the highest coefficient for D_2, D_3 and D_4 and the second one for D_1 and therefore, it can be considered as the most relevant one for the problem.[3]
- Electricity consumption of households (ELECT): This variable presents an important impact on all the models but model D_3. This may be because this variable takes very dispersed values for this class, making it not very discriminative for the subproblem. Note that the coefficient of this variable for model D_4 is very high, precisely because $\mathcal{C}_4$ is characterised by very low

Table 5
Results obtained (*Acc* and *MMAE* for the test sets) for different data partitions and the TEOC (LR) method.

Experiment	*Acc*	*MMAE*
Training 2005–2007 testing 2008	74.074	0.8571
Training 2008–2009 testing 2010	85.185	0.8000
Training 2005–2009 testing 2010	92.593	0.2857

Table 6
Projection coefficients $\mathbf{w}_i$ for each variable and the different decisors.

	D_1	D_2	D_3	D_4
Coefficients				
GDPGR	0.88	−1.61	−0.59	−0.53
RLPGH	−7.90	−9.04	−24.07	−26.75
EMPLO	−1.98	2.33	1.27	0.78
RDEXPE	−2.04	−0.57	1.35	1.76
RESPROD	3.38	3.27	7.56	7.79
ORGAN	−2.32	−3.54	1.70	2.81
ELECT	−10.85	−6.12	−0.71	−14.70
RISKPOV	−1.64	−1.31	0.56	1.04
EARLY	0.93	1.67	0.96	1.34
EMPOLD	−2.62	−5.91	−4.56	−5.71
PUBDE	0.32	−1.53	−3.91	−3.63
LIFEE	6.09	2.98	4.20	5.49
HELIF	−0.73	−2.18	−1.73	−1.02
GREGA	−3.43	−2.35	3.26	2.42
RENWA	−0.07	−1.75	−3.05	−4.32
TRANS	−1.72	−1.66	1.34	0.65
GGTRA	10.74	5.07	−2.12	11.63
ASSIS	0.72	−2.76	5.03	8.77
EUIMP	−6.25	−4.94	−3.58	−6.19
Thresholds				
b_1	−4.49	−7.40	6.12	11.76
b_2		1.76	12.47	

Table 7
Ranking of variables according to the impact on each model $|\mathbf{w}_i|$.

D_1	D_2	D_3	D_4
ELECT	RLPGH	RLPGH	RLPGH
GGTRA	ELECT	RESPROD	ELECT
RLPGH	EMPOLD	ASSIS	GGTRA
EUIMP	GGTRA	EMPOLD	ASSIS
LIFEE	EUIMP	LIFEE	RESPROD
GREGA	ORGAN	PUBDE	EUIMP
RESPROD	RESPROD	EUIMP	EMPOLD
EMPOLD	LIFEE	GREGA	LIFEE
ORGAN	ASSIS	RENWA	RENWA
RDEXPE	GREGA	GGTRA	PUBDE
EMPLO	EMPLO	HELIF	ORGAN
TRANS	HELIF	ORGAN	GREGA
RISKPOV	RENWA	RDEXPE	RDEXPE
EARLY	EARLY	TRANS	EARLY
GDPGR	TRANS	EMPLO	RISKPOV
HELIF	GDPGR	EARLY	HELIF
ASSIS	PUBDE	ELECT	EMPLO
PUBDE	RISKPOV	GDPGR	TRANS
RENWA	RDEXPE	RISKPOV	GDPGR

[3] Following the SD Report 2011, this indicator must be interpreted with caution. The comparability between countries is hampered by methodological changes and also by the cultural effect and the peculiarities of translated questions [13].

values of this variable (given the better economic activity and the lower use of electronic devices), thus being very useful for the subproblem.

- Transport greenhouse emissions (GGTRA): This variable can be said to be considerably influential for most of the subproblems (except for D_3 as well, where it also takes dispersed values).

Furthermore, if the scenarios are separately analysed (see Table A.8 for a description of the variables of each scenario), the impact of each one in the classification problem can be studied. From this analysis, it can be said that one of the most important scenarios is the sustainable consumption and production (most of the variables seem to have a high impact in most of the models), followed by the demographic changes, the global partnership and the climate change and energy scenarios. The scenarios that can be said to present a low impact on most of the models are public health (except the LIFEE variable), sustainable transport (except the GGTRA variable) and social inclusion. Finally, it can be argued that except for the RLPGH variable (that has been characterised before as the most influential in most of the models), the socioeconomic development scenario has a very low impact in the majority of the models.

For concluding this model analysis, some conclusions can be drawn from the coefficient sign of some of the variables. Note that

Table A.8
Variables definition by scenario.

Variable	Description
Scenario 1: socioeconomic development	
GDPGR (Key indicator): real growth of gross domestic product (GDP) per capita (Ratio)	This indicator defines percentage change GDP in real terms per inhabitant in reference year in comparison with previous year. Real GDP per capita is calculated as the ratio of annual value of gross domestic product at constant prices to the average population of country (territorial units)
RLPGH: Labour productivity per hour worked (Ratio)	Calculated as real output (deflated GDP measured in chain-linked volumes, reference year 2005) per unit of labour input (measured by the total number of hours worked)
EMPLO: total employment rate (%)	Calculated by dividing the number of employed persons aged from 20 to 64 by the total population of the same age group
RDEXP: R&D total expenditure (%)	The indicator provided is GERD (Gross Domestic Expenditure on Research and Development) as a percentage of GDP
Scenario 2: sustainable consumption and production	
RESPROD (Key indicator): resource productivity (Ratio)	Resource productivity is GDP divided by domestic material consumption (DMC). DMC measures the total amount of materials directly used by an economy
ORGAN: area under organic farming (%)	The share of total utilised agricultural area occupied by organic farming (existing organically-farmed areas and areas in process of conversion)
ELECT: Electricity consumption of households (1000 tonnes of equivalence)	Defined as the quantity of electricity consumed by households. Household consumption covers all use of electricity for space and water heating and all electrical appliances
Scenario 3: social inclusion	
RISKP (Key indicator): people at risk of poverty or social exclusion (%)	The sum of persons who are at risk of poverty or severely materially deprived or living in households with very low work intensity (at risk-of-poverty are persons with an equivalised disposable income below the 60% of the national median equivalised disposable income (after social transfers)
EARLY: early leavers for education and training (%)	It refers to persons aged from 18 to 24 fulfilling the following conditions: first, that the highest level of education or training attained is ISCED 0, 1, 2 or 3c short, and second, the respondents declare not having received any education or training in the four weeks preceding the survey
Scenario 4: demographic changes	
EMPOLD (Key indicator): employment rate of older workers (%)	Calculated by dividing the number of employed persons aged from 55 to 64 by the total population of the same age group
PUBDE: general government gross debt (%)	Defined (in the Maastricht Treaty) as consolidated general government gross debt at nominal value, outstanding at the end of the year
Scenario 5: public health	
LIFEE: Life expectancy at age 65 (years)	The average number of years still to be lived by a person who has reached the age 65, if subjected throughout the rest of persons life to the current mortality conditions
HELIF (Key indicator): healthy life years and life expectancy at birth (years)	The number of years that a person at birth is still expected to live in a healthy condition, combining information on mortality and morbidity
Scenario 6: climate change and energy	
GREGA: greenhouse gas emissions (Index, base year = 100)	The share of greenhouse gas emissions (carbon dioxide, methane and nitrous oxide; total man-made emissions of the "Kyoto basket" of greenhouse gases) in the gross inland energy consumption in relation to based year 2000 = 100
RENWA (Key indicator): share of renewable energy in gross final energy consumption (Ratio)	Calculated as the share of renewable energy in gross final energy consumption. Renewable energy sources is energy originating from natural, repeatable natural processes, mainly energy generated from solar radiation, wind, water, geothermal resources, biomass, biogas and liquid biofuels
Scenario 7: sustainable transport	
TRANS (Key indicator): energy consumption of transport relative to GDP (Index, year 2000 = 100)	Ratio between the energy consumption of transport and GDP (chain-linked volumes, at 2000 exchange rates). The energy consumed by all types of transport is covered, including commercial, individual and public transport, with the exception of maritime and pipeline transport
GGTRA (Key indicator): greenhouse gas emission from transport 1000 tonnes of CO_2 equivalent)	This indicator shows trends in the emissions from transport (road, rail, inland navigation and domestic aviation) of the greenhouse gases regulated by the Kyoto Protocol. Only three gases are relevant in the context of transport (carbon dioxide, methane, and nitrous oxide) and these have been aggregated according to their relative global warming potentials
Scenario 8: global partnership	
ASSIS (Key indicator): Official development assistance as share of gross national income (%)	Official development assistance (ODA) consists of grants or loans that are undertaken by the official sector with promotion of economic development and welfare in the recipient countries as the main objective
EUIMP: EU Imports from developing countries by income group (Billion eur).	The value at market prices of EU imports from the Development Assistance Committee countries, as they have been determined by this Committee

this sign indicates if the variable is positively or negatively correlated with a greater SD (e.g., recall that if the classes are presented ordered in the projections in an ascent fashion a negative sign for a given coefficient means that the higher the value for that variable, the higher the probability of belonging to one of the first classes, i.e., the ones with better SD). Because of that, by analysing the coefficients in Table 6, it can be said that the higher the value of the variables RLPGH, ELECT, EMPOLD, HELIF, RENWA and EUIMP is, the higher the probability of belonging to a class presenting a high SD will be. These variables are the ones that present a negative coefficient for all of the models. On the contrary, the higher the value of the variables RESPROD, EARLY and LIFEE[2] is, the higher the probability of belonging to classes with a lower SD will be.

Concerning the variables with different sign for different models, this indicates that they do not follow a stable increase or decrease with respect to the class ordering.

4. Conclusions

Sustainable development (SD) is a major global trend in the international political debate in the current global context. Policy-makers are called upon to assess the impact of their strategies in terms of SD and for this task, they should be provided with models, tools, indicators or even rankings for evaluating the progress of nations towards SD in their three pillars: environmental, economic and social. However, there is a lack of predictive models in this context, that is, models able to rank countries according to their SD advances and to generate information on which future actions can be based. This information would allow policy-makers to support policy development and monitor the effects of policy responses.

In this sense, one of the main objectives of this paper is to perform a study of the progress towards SD of the 27 EU countries and to analyse the predictive capability of machine learning methods in this context. To do so, a clustering analysis was performed to obtain a set of clusters that group countries with similar SD major indicators values. The clusters obtained were ordered according to their overall SD performance (the clusters were categorised as advanced, followers, moderate and initiated). The characterisation of the clusters obtained reflects a global picture of the SD stage of countries, which could enrich and complement the judgement of stakeholders more than a single indicator score value or trying to find the SD readiness of a country through separate indicators.

According to the inherent order of the labelling space, a new ordinal regression algorithm is proposed, which decomposes the original problem into several subproblems and joins the different outputs using a class conscious decision function. By this simplification, the base methodologies can be significantly improved and more simple and interpretable base methodologies can be used at a similar performance. The ordinal regression proposed algorithm is compared to other related classifiers and shows to be competitive yielding better results for this application and supporting the initial assumption of the ordinal nature of clusters defined by the expert and the clustering algorithm.

For this study, Logistic Regression was one of the methods used as the base methodology for the ensemble, and thus, the most determinant variables for the target label can be studied. These variables are the labour productivity per hour worked, the electricity consumption of households and the transport greenhouse emissions. In regards to the scenarios, the most important are sustainable consumption, demographic changes, global partnership and climate change and energy, a result that is in line with the three dimensions of SD.

Although it is difficult to assess the direct impact of the indicators on the progress towards sustainability, it can be stated that given the good generalisation performance of the methodology, it may be useful to monitor national strategies for European governments, in a manner similar to that used for rankings (because of the ordinal nature of the clusters obtained), as a managerial tool for supporting decision making and for benchmarking practices to compare results.

In addition, it can be said that as appropriate data become available, the implications of our approach would appear to be worthwhile for future comparative-research in cross-country performance. In future lines of research, it will be of interest to amplify the study with data since the year 2011 to capture the whole impact of the economic and financial crisis and its evolution. Finally, although the variables selected for this study were the official ones, they may bias both the results of the clustering analysis and the classification methods. Therefore, the selection of alternative, additional and/or a combination of variables for each scenario is proposed for further research.

Acknowledgment

This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P2011-TIC-7508 project of the "Junta de Andalucía" (Spain).

Appendix A. Variables used for the study

Table A.8 shows a description of the variables used for the clustering analysis and the learning process. The first column of the table shows the complete and abbreviated name of the variable, whether it is considered a key indicator or not and unit of measurement.

References

[1] A. Agresti, Categorical Data Analysis, second ed., Wiley Series in Probability and Statistics, Wiley-Interscience, 2002.
[2] F. Allievi, J. Luukkanen, J. Panula-Ontto, J. Vehmas, Grouping and ranking the eu-27 countries by their sustainability performance measured by the eurostat sustainability indicators, in: Trends and future of sustainable development, 2011, pp. 9–20.
[3] A. Alshami, A. Lotfi, S. Coleman, Unified knowledge based economy neural forecasting map, in: The 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, 2012, pp. 1–8.
[4] K.J. Arrow, P. Dasgupta, L.H. Goulder, K.J. Mumford, K. Oleson, Sustainability and the measurement of wealth, Environ. Dev. Econ. 17 (2012) 317–353.
[5] S. Baccianella, A. Esuli, F. Sebastiani, Evaluation measures for ordinal regression, in: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 09), Pisa, Italy, 2009, pp. 283–287.
[6] P.M. Boulanger, T. Brechet, Models for policy-making in sustainable development: the state of the art and perspectives for research, Ecol. Econ. 55 (2005) 337–350.
[7] W. Chu, S.S. Keerthi, Support vector ordinal regression, Neural Comput. 19 (2007) 792–815.
[8] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, first ed., Cambridge University, 2000.
[9] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutiérrez, A preliminary study of ordinal metrics to guide a multi-objective evolutionary algorithm, in: 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011), 2011.
[10] A.L. Dahl, Achievements and gaps in indicators for sustainability, Ecol. Indic. 17 (2012) 14–19.
[11] J.A. Du Pisani, Sustainable development historical roots of the concept, Environ. Sci. 3 (2006) 83–96.
[12] European Commission, A sustainable europe for a better world: a European Union strategy for sustainable development, European Commission's proposal to the Gothenburg European Council, 2001.
[13] Eurostat, Sustainable development in the European Union: 2011 monitoring report of the eu sustainable development strategy, Office for Official Publications of the European Communities, 2011.
[14] P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, C. Hervás-Martínez, An experimental study of different ordinal regression

methods and measures, in: 7th International Conference on Hybrid Artificial Intelligence Systems (HAIS), 2012, pp. 296–307.
[15] J. Hall, E. Giovannini, A. Morrone, G. Ranuzzi, A framework to measure the progress of societies, OECD Statistics Working Papers 2010/5, OECD Publishing, 2010.
[16] R. Herbrich, T. Graepel, K. Obermayer, Support vector learning for ordinal regression, in: International Conference on Artificial Neural Networks, 1999, pp. 97–102.
[17] J. Stiglitz, A. Sen, J.P. F., Report by the commission on the measurement of economic performance and social progress, 2009. <http://www.stiglitz-sen-fitoussi.fr/documents/rapport_anglais.pdf>.
[18] M. Jacobs, Fairness and Futurity: Essays on Environmental Sustainability and Social Justice, Oxford University Press, 1998. Chapter Sustainable Development as a Contested Concept.
[19] A. Kulig, H. Kolfoort, R. Hoekstra, The case for the hybrid capital approach for the measurement of the welfare and sustainability, Ecol. Indic. 10 (2010) 118–128.
[20] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[21] N. Lee, C. Kirkpatrick, Sustainable development and integrated appraisal in a developing world, Sustainable Development and Integrated Appraisal in a Developing World, Edward Elgar Pub., 2000.
[22] H.Y. Lin, Feature selection based on cluster and variability analyses for ordinal multi-class classification problems, Knowl.-Based Syst. 37 (2013) 94–104.
[23] T. Luzzati, G. Gucciardi, Comparing the sustainability of the 27 European countries. a robustness approach, in: 10th International Conference of the European Society for Ecological Economics (ESEE 2013), 2013, p. 453.
[24] P. McCullagh, Regression models for ordinal data, J. Roy. Stat. Soc. 42 (1980) 109–142.
[25] D. Meadows, D. Meadows, J. Randers, W. Behrens, The limits to growth: a report for the Club of Rome's project on the predicament of mankind. A Potomac Associates Book, Universe Books, 1974.
[26] S. Menard, Logistic Regression: From Introductory to Advanced Concepts and Applications, SAGE Publications, 2009.
[27] B. Moldan, S. Janoušková, T. Hák, How to understand and measure environmental sustainability: indicators and targets, Ecol. Indic. 17 (2012) 4–13.
[28] M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, Projection based ensemble learning for ordinal regression, IEEE Trans. Cybernet. 44 (5) (2014) 681–694.
[29] J.C.V. Pezzey, Sustainable Development Concepts; An Economic Analysis. Working Papers. World Bank – The World Bank Environment Paper, 1992.
[30] L. Pintér, P. Hardi, P. Bartelmus, Sustainable development indicators: proposals for the way forward prepared for the United Nations Division for Sustainable Development, International Institute for Sustainable Development, 2005.
[31] E. Rametsteiner, H. Pulzl, J. Alkan-Olsson, P. Frederiksen, Sustainability indicator development science or political negotiation, Ecol. Indic. 11 (2011) 61–70.
[32] D.S. Rogers, A.K. Duraiappah, D.C. Antons, P. Munoz, X. Bai, M. Fragkias, H. Gutscher, A vision for human well-being: transition to social sustainability, Curr. Opin. Environ. Sust. 4 (2012) 61–73.
[33] M. Rojas, The measurement of economic performance and social progress report and quality of life: moving forward, Soc. Indic. Res. 102 (2011) 169–180.
[34] A. Shashua, Levin, Advances in neural information processing systems, Ranking with Large Margin Principle: Two Approaches, vol. 15, MIT Press, Cambridge, 2003. pp. 937–944.
[35] A.J. Smola, B. Schölkopf, K.R. Müller, The connection between regularization operators and support vector kernels, Neural Netw. 11 (1998) 637–649.
[36] UNCED, Agenda 21: The United Nations Program of Action from Rio. Technical Report. United Nations Conference in Environmental Development, 1992.
[37] UNDP, Sustainability and Equity: A Better Future for All. Human Development Report. Technical Report. United Nations Development Program, 2011.
[38] WCED, Report of the world commission on environment and development: Our common future, 1997. <http://www.un-documents.net/wced-ocf.htm>.

3.3. An organ allocation system for liver transplantation based on ordinal regression

Nowadays, liver transplantation is a widely-accepted treatment for patients with a terminal liver disease. However, it is well-known that transplantation is greatly hampered by the unavailability of suitable liver donors; several methods have been then developed and applied to find a better system to prioritise recipients on the waiting list. Most of these methods only consider donor or recipient characteristics separately, not considering the potential compatibility of the organ and not respecting then the principles of fairness and benefit. This could lead to a risk of unconscious gaming when trying to match marginal donors to urgent candidates.

To solve these deficiencies, this paper proposes a novel donor-recipient allocation system for liver transplantation. The dataset is comprised of donor-recipient pairs from different centres (seven Spanish and one United Kingdom hospitals). The problem is assessed from an ordinal classification point of view due to the natural order of the classes (failure of the organ before 15 days after transplantation, failure between 15 days and three months, failure of the organ between three months and one year, and no failure of the organ presented up to more than one year after transplantation). The classification method makes use of a cascade binary decomposition method specially designed to handle the imbalanced nature of the data and the error correcting output codes strategy for fusing the output of the different models [34]. The best model obtained is used in conjunction with the model for end-stage liver disease score (MELD) [64], which estimates the patients severity and is one of the most used current assignation methodologies.

The experiments on this paper show that the methodology presented is competitive for all the metrics selected when compared to other machine learning techniques and efficiently complements the MELD score based on the principles of efficiency and equity (helping to avoid draws in most cases).

A simulation of the proposed system is also included, in order to visualise its behaviour in more realistic situations. This experiment has shown that there are some determining factors in the characterization of the survival time after transplantation (concerning both donors and recipients) and that the joint use of these sets of information could be, in fact, more useful and beneficial for the survival principle. Nonetheless, the results obtained indicate as well the true complexity of the problem and the fact that other characteristics that have not been included in the dataset may be of importance for the characterization of the dependent variable (survival time after transplantation), thus starting a promising line of future work.

Applied Soft Computing 14 (2014) 88–98

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

An organ allocation system for liver transplantation based on ordinal regression

M. Pérez-Ortiz [a,*], M. Cruz-Ramírez [a], M.D. Ayllón-Terán [b], N. Heaton [c], R. Ciria [b], C. Hervás-Martínez [a]

[a] Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
[b] Liver Transplantation Unit, Reina Sofía Hospital, Córdoba, Spain
[c] Liver Transplantation Unit, King's College, London, United Kingdom

ARTICLE INFO

Article history:
Received 1 February 2013
Received in revised form 23 July 2013
Accepted 26 July 2013
Available online 5 September 2013

Keywords:
Liver transplantation
Survival analysis
Machine learning
Support vector machines
Ordinal regression
Decision-making

ABSTRACT

Liver transplantation is nowadays a widely-accepted treatment for patients who present a terminal liver disease. Nevertheless, transplantation is greatly hampered by the un-availability of suitable liver donors; several methods have been developed and applied to find a better system to prioritize recipients on the waiting list, although most of them only consider donor or recipient characteristics (but not both). This paper proposes a novel donor–recipient liver allocation system constructed to predict graft survival after transplantation by means of a dataset comprised of donor–recipient pairs from different centres (seven Spanish and one UK hospitals). The best model obtained is used in conjunction with the Model for End-stage Liver Disease score (MELD), one of the current assignation methodology most used globally. This problem is assessed using the ordinal regression learning paradigm due to the natural ordering in the classes of the problem, via a cascade binary decomposition methodology and the Support Vector Machine methodology. The methodology proposed has shown competitiveness in all the metrics selected, when compared to other machine learning techniques and efficiently complements the MELD score based on the principles of efficiency and equity. Finally, a simulation of the proposal is included, in order to visualize its performance in realistic situations. This simulation has shown that there are some determining factors in the characterization of the survival time after transplantation (concerning both donors and recipients) and that the joint use of these sets of information could be, in fact, more useful and beneficial for the survival principle. Nonetheless, the results obtained indicate the true complexity of the problem dealt within this study and the fact that other characteristics that have not been included in the dataset may be of importance for the characterization of the dependent variable (survival time after transplantation), thus starting a promising line of future work.

1. Introduction

During the last few decades, new trends in biomedicine have considered using some machine learning techniques as classification methods [1,2], which has worked well in a great number of problems and resulted in remarkable applications for science [3,4]. Liver transplantation is an accepted treatment for patients who present end-stage liver disease. However, transplantation is restricted by the lack of suitable liver donors; this imbalance between supply and demand resulting in significant waiting list death. In order to cope with this situation, several methods have been developed and applied to find a better system to prioritize recipients on the waiting list.

The first attempt at developing a system was the Donor Risk Index (DRI) [5], aimed at establishing the quantitative risk associated with the transplant when considering donor information. Another widely validated methodology that is the cornerstone of current allocation policy, is the Model for End-stage Liver Disease (MELD) [6], which is based on the "sickest-first" principle, where the only aspect considered is information concerning the recipient. The use of expanded criteria donors (donors with extreme values of age, days in the intensive care unit (ICU), inotrope usage, body mass index (BMI) and cold ischemia time) results in an increased risk of recipient and/or graft losses compared to the risk associated with the use of livers from non-extended criteria donors [7]. These risks should be carefully analysed since the combination of several of these risk factors can result in graft loss [8]. Nevertheless, these

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain. Tel.: +34 957 218 349; fax: +34 957 218 630.
E-mail addresses: i82perom@uco.es (M. Pérez-Ortiz), mcruz@uco.es (M. Cruz-Ramírez), lolesat83@hotmail.com (M.D. Ayllón-Terán), nigel.heaton@nhs.net (N. Heaton), rubenciria@hotmail.com (R. Ciria), chervas@uco.es (C. Hervás-Martínez).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.07.017

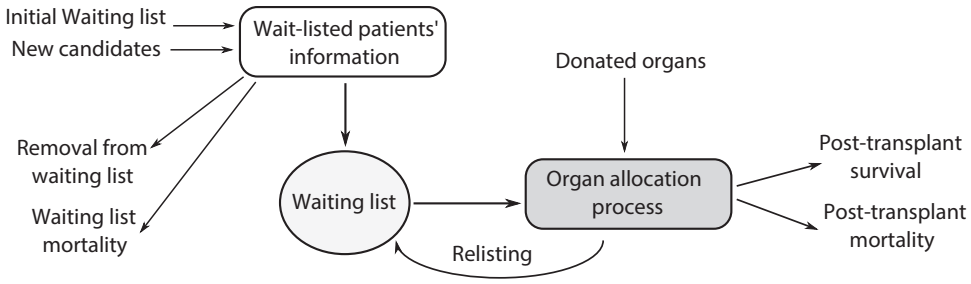


Fig. 1. Graphic representing the organ allocation process.

methods can not be considered good predictors of graft failure after transplantation since they only take into consideration either characteristics of donors or of recipients (but not both), when there could actually be more complex factors involved in the situation (donor, recipient and transplant organ characteristics). In order to deal with this problem, Rana et al. [9] devised a scoring system (SOFT) that predicted recipient survival 3 months after liver transplantation, which is intended to complement MELD-predicted waiting list mortality rates by making use of both donor and recipient characteristics. P. Dutkowski et al. recently proposed a balance of risk (BAR) score [10] based on donor and recipient characteristics. A rule-based system was used to determine graft survival 1 year after the transplant [11]. The input of this rules-based system being the response of two artificial neural networks trained with donor, recipient and transplant organ characteristics.

Fig. 1 graphically represents the process of organ allocation (figure restructured from [12]). Generally, donors are assigned to the candidates under the greatest-risk according to the MELD score. This policy does not allow the liver transplant team to match the donor to the recipient according to principles of fairness and benefit. This could lead to a risk of unconscious gaming when trying to match marginal donors to urgent candidates.

In the same vein, this paper considers a liver transplant dataset obtained from seven different Spanish hospitals and King's College Hospital in the UK and includes characteristics of donors, recipients and transplant organs, with the aim of developing and constructing a supranational system for predicting graft survival, by means of intelligent classification techniques. Although this problem has been tackled successfully before by means of a binary classification task [11], a significant contribution of this paper is that the classification problem is addressed using an ordinal regression point of view since the classes are ordered taking into account the time leading up to liver failure (in case of failure) providing therefore more information about the hypothetical graft failure. The classification problem could be also tackled by a multiclass classification problem but this approach will ignore the ordering information present in the output space. The classes involved in the dataset are: (1) failure of the graft within the first 15 days after transplantation, (2) failure between 15 days and 3 months, (3) failure between 3 months and 1 year, and (4) no failure presented. These intervals have been highlighted by experts as being the most pertinent in early graft loss. Several issues need to be taken into account in order to exploit the presence of this order structure. First of all, the learner (classifier, in this case) could benefit from this implicit ordering in order to construct more robust and fairer decision regions for the data, since the classification errors to be minimized vary from the ones considered in the nominal classification paradigm (the zero-one loss function). Secondly, with the final aim of evaluating the performance of those classifiers, different measures or metrics could be developed and used.

In order to clarify the differences among these paradigms, the problem of classifying tumour cells given the labels: {*normal cell*, *dysplastic cell*, *tumor cell*, *metastatic cell*} could be considered. Clearly, an order among the categories can be appreciated, and there are also some misclassification errors that should be more penalized. For example, misclassifying a *metastasis cell* with a *normal cell* should be far more penalized than misclassifying it as a *tumor cell*. Since this is a common learning issue, several approaches to tackle this paradigm (known as ordinal regression or ordinal classification) have been proposed in the domain of pattern recognition and machine learning over the years, ever since the first work applying logistic regression dating back to 1980 [13]. This issue has generally been addressed by transforming ordinal scales into numeric values and solving the problem as one of standard regression or multinomial classification. However, there are several problems within this approach: on the one hand, the fact that, without a priori knowledge, the distance between different classes is unknown, thus the assumption of equidistant labels when performing standard regression may not hold; on the other hand, as nominal classification does not consider this order information, misclassification errors are treated equally. Nonetheless, other works have approached the paradigm considering the order information by means of threshold methods [14–16] which are based on the idea that some underlying real-valued outcomes exist (also called latent variable), although they are unobservable.

However, there is still a major group of classification techniques specially designed for approaching ordinal regression which are based on the idea of decomposing the original problem into a set of binary classification tasks [17,18], or by formulating the original problem as one of extended binary classification [19,20]. Each subproblem can be solved in this case either by a single model or by a multiple set. The subproblems are defined in this case by a very natural methodology, considering whether a pattern $\mathbf{x}$ belongs to a class greater than a fixed k [21], and finally combining the binary predictions into a unique ordinal label. The idea of decomposing the target variable in simpler classification tasks has demonstrated to be very powerful in the context of ordinal regression.

In this paper, due to the complexity of the classification problem, which presents an ordinal and highly unbalanced nature leading to difficulties in the classification of the three minority categories (note that from class 1 to 4 the number of patterns per class are respectively {76, 76, 62, 1223}), a binary decomposition method for ordinal regression known as *OneVsPrevious* is considered by using the well-known Support Vector Machine classifier (SVM) and two different approaches to combine the classifier outputs. Note that, although other base methodologies could be used for the decomposition method, such as artificial neural networks, the SVM paradigm has been chosen in order to provide a fair comparison in the experiments performed since most of the methods proposed for ordinal classification are based on SVMs [16,19,18]. Therefore, a SVM model is created for each subproblem by solving a global optimization problem seeking for the optimal separating hyperplane for the data and optimizing the parameters using a nested cross-validation over the parameters space. The methodology, which shows competitive results when compared to other ordinal and nominal approaches (based on

SVM [16,19], artificial neural networks [29] and logistic regression [15]), is then used to develop a complete system of liver allocation, in conjunction with the MELD system known worldwide.

The paper is organized as follows: Section 2 shows a description of the ordinal regression methodology used in this work; Section 3 thoroughly explains the constructed dataset and the experiments to be performed; Section 4 presents and analyses the results of the above-mentioned experiments. In Section 5, a simulation of the proposal is performed and, finally, Section 6 outlines some conclusions and future work.

2. Methodology

This section establishes the terminology and notation that will be used throughout the entire work, as well as the ordinal regression method used. The goal in classification is to assign an input vector $\mathbf{x}$ to one of K discrete classes $\mathcal{C}_k$, where $k \in \{1, \ldots, K\}$. A formal framework for the ordinal regression problem could be introduced by considering an input space $\mathcal{X} \in \mathbb{R}^d$, where d is the data dimensionality. To do so, an outcome space $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$ is defined, where the labels are ordered due to the data ranking structure (i.e. $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$, where $\prec$ denotes this order information) thus being $\mathcal{Y}$ a non-metric space. Let N be the number of patterns in the sample and N_k the number of samples for the kth class. The objective in this kind of problem is to find a prediction function $f : \mathcal{X} \to \mathcal{Y}$ by using an i.i.d. sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$.

2.1. Base methodology

The Support Vector Machine paradigm (SVM) [22,23] is perhaps the most common kernel learning method for statistical pattern recognition due to its good generalization ability and freedom from local minima. The basic idea behind this technique is the separation of two different classes through a hyperplane which is specified by a normal vector $\mathbf{w}$ and a bias b. The optimal separating hyperplane is the one which maximizes the distance between the hyperplane and the nearest points in both classes (called margin). Beyond the application of kernel techniques to allow non-linear decision discriminants (the kernel trick), another generalization was made to replace hard margins with soft margins [23], using the so-called slack-variables ξ_i in order to avoid inseparability, relax the constraints and handle noisy data. Therefore, this algorithm seeks a classifier $f : \mathbb{R}^d \to \mathbb{R}$ of the form $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ (Φ being the mapping function induced by the kernel) that minimizes the objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i, \tag{1}$$

for some parameter C and subject to the constraints:

$$y_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \qquad \forall_i \in \{1, \ldots, N\}.$$

2.2. Binary decomposition

The binary decomposition method known as the cascade linear utility model [25] is used in this case. This procedure considers $K-1$ models $\mathcal{D}$ (each model D_i will be comprised in this case of a projection $\mathbf{w}$ and a threshold b_i), in such a way that model k separates classes $\mathcal{C}_1 \vee \cdots \vee \mathcal{C}_{K-k-1}$ from class $\mathcal{C}_{K-k}$, so that not all the classes are considered in the computation of each model (as can be seen in Fig. 2, where the decomposition is described graphically). This methodology was mainly used to balance some projections, due to the highly unbalanced character of the liver transplantation sample constructed in this paper, which leads to the misclassification of minority classes for the sake of good overall performance, which in this case can be achieved with a trivial classifier that always predicts the majority category. It is also noteworthy that the classes could be inversely decomposed by making use of the *OneVsFollowers* decomposition [26], although, in this situation, it would not be advisable because the class imbalance will increase, since the first classes are the most unbalanced ones.

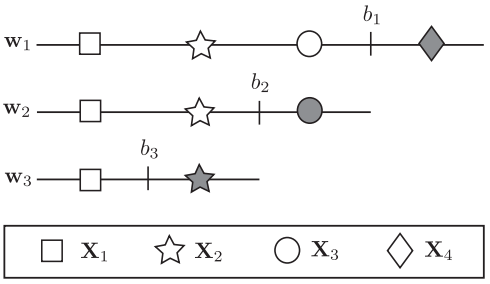


Fig. 2. Binary decompositions performed for a 4-category dataset, where $\mathbf{X}_k$ is defined as the patterns belonging to class k, $\mathbf{w}_i$ represents the ith projection and b_i the bias associated with that projection. White-shadowed figures represent the negative class, and black-shadowed ones the positive class, while the shape represents the original category.

The training set for model or decision maker $D_k = \{\mathbf{w}_k, b_k\}$ is specified by $\{\mathbf{X}_{(i|i<k)}, \mathbf{X}_{(i|i=k)}\}$. Therefore, a coding matrix $\mathbf{M}_{(K-1\times K)}$ associated to the $K-1$ binary decompositions of the cascade utility model can be defined as follows:

$$\mathbf{M} = \begin{pmatrix} -1 & -1 & -1 & +1 \\ -1 & -1 & +1 & 0 \\ -1 & +1 & 0 & 0 \end{pmatrix},$$

where the label -1 is assigned to patterns corresponding to the negative class, the label +1 to patterns belonging to the positive class, and finally, the patterns associated with label 0 are excluded for the training process. This matrix can be obtained by means of a single model (by using neural networks for example) or by a multiple set of models (training a binary classifier for each subproblem, as in this paper). Thus, once the model or models have been trained, a set of $K-1$ decision values $\mathbf{f}(\mathbf{x})$ are obtained for pattern $\mathbf{x}$. At that point, and concerning the prediction phase, two different alternatives have been considered in this work:

- Hierarchical approach: This method was originally proposed for prediction in the cascade utility model [25], which operates in a forward manner, starting the prediction phase with the first model and going forward, until the model predicts one class that is not decomposed in the following models.
- Error-Correcting Output Codes framework (ECOC): Although the hierarchical approach may work well in a great number of problems, prediction depends on the projections evaluated in the first stage, and is thus biased towards those classes. Because of that, a different proposal to be considered in the prediction phase is also evaluated in this paper. Based on these concepts, many efforts have been made by the machine learning community in order to reformulate binary classifiers to the multinomial case, resulting in methods such as the *OneVsOne* or *OneVsAll* paradigms. Nonetheless, other proposals can also be found in the state-of-the-art literature, such as the Error-Correcting Output Codes methodology [27]. The principal idea is to associate each class $k \in \mathcal{Y}$ with a column of a binary coding matrix $\mathbf{M} \in \{-1, +1\}^{l\times K}$ for a given l (note that in our case l will be preassigned to $K-1$, i.e. the number of binary decompositions performed). The binary algorithm is run once for each row of the matrix in the induced binary classification problem, f yielding as many hypotheses as l. Prediction is then accomplished by choosing the column of $\mathbf{M}$ closest to the set of decision values $\mathbf{f}(\cdot) = f_1(\cdot), \ldots, f_l(\cdot)$. A slightly

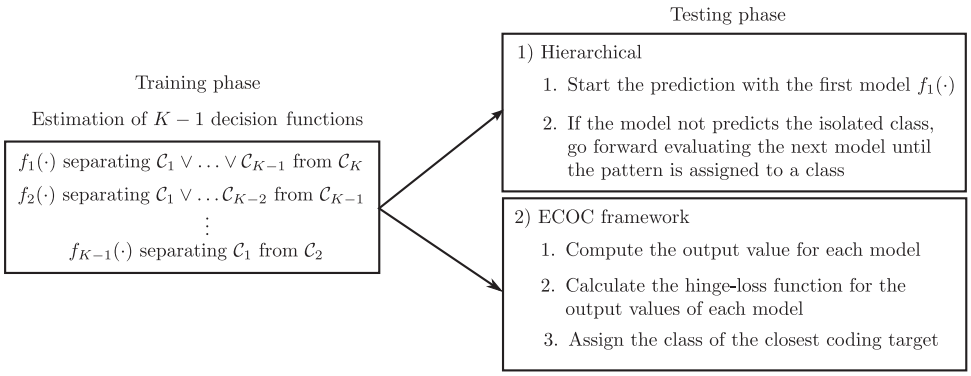


Fig. 3. Summarization of the two testing proposals for the cascade utility training model.

modified version of this proposal is given in [28], where the coding matrix is taken instead from the set $\mathbf{M} \in \{-1, 0, +1\}^{l \times K}$ (as the one previously defined for our decomposition problem), leading to an indifferent condition in the prediction phase for patterns with label 0. The main handicap within this paradigm is the choice of a suitable loss function for the binary classifier chosen as the base method. In this case, due to the choice of the 1-norm SVM paradigm as the base method, the hinge-loss function is chosen, which is the most commonly used for SVM. In order to see this, let us formulate Eq. (1) in terms of the error:

$$\underbrace{\frac{1}{2}||\mathbf{w}||^2}_{\text{minimizer}} + \underbrace{C\sum_{i=1}^{N} loss(y_i, f(\mathbf{x}_i))}_{\text{error}},$$

where the error function chosen is usually the hinge-loss (for L1-SVM), or its square (for L2-SVM):

$$H(\mathbf{x}_i) = (1 - y_i \cdot f(\mathbf{x}_i))_+ = \max(0, 1 - y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b)) = \xi_i.$$

The different possibilities for $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b)$ are:
- $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) > 1$: the point is well-classified and outside the margin.
- $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) = 1$: the point is on the margin.
- $y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) < 1$: the point is within the margin or misclassified.

Hence, one of the main advantages of this methodology is that real values for prediction are used instead of binary predicted class values; thus the model will be provided with additional information which may be useful for improving its performance. Indeed, the real values used are related with the distance to the threshold, a measure usually used in this cases for estimating the probability of belonging to a class.

With comparison and clarification purposes, Fig. 3 presents a summary of the two proposals of the paper.

3. Experiments performed

In this section, a complete description of the multi-state liver transplantation dataset is given, followed by the methods to be compared and the measures evaluated.

3.1. Dataset description

First of all, a multi-centred retrospective analysis was made of 7 Spanish Liver Transplant units. Recipient and donor characteristics were reported at the time of transplant. Patients undergoing partial, split or living-donor liver transplantation and patients undergoing combined or multivisceral transplants were excluded from the study. All patients were followed from the date of transplant until either death, or graft loss prior to 1 year after transplantation. Liver Transplantation units were homogeneously distributed throughout Spain. The dataset constructed has 634 patterns (donor–recipient pairs) corresponding to the years 2007 and 2008. In addition, the dataset was completed with information about donor–recipient pairs from the King's College Hospital (London), to perform a supranational study of donor–recipient allocation in liver transplantation. To obtain a similar number of patterns, only reported pairs of recipients over 18 years of age between January 2002 and December 2010 were included. Thus, a dataset containing 858 English donor–recipient pairs were collected. In order to merge the datasets, several variables were selected, 16 recipient variables, 17 donor variables and 5 surgically related variables. Furthermore, all patients were followed from the date of transplant until death, graft loss or completion of the first year after the liver transplant.

Once the data was collected, it was necessary to perform some classical techniques of data imputation in order to replace all the missing values. To do so, first, when the ratio of missing values for any variable was under 1%, these were substituted by the mean (in the case of a continuous and quantitative variable) and by the mode (in the case of a binary and qualitative one). When the ratio of missing values was over 1% and under 10%, a linear and non-linear regression analysis was performed to recover the missing values. Finally, patterns with over 10% of missing values were not considered for the study; 19 and 36 patterns were discarded after applying the techniques of data imputation, leaving a total of 615 Spanish patterns and 822 English patterns, respectively. Therefore, the resulting dataset was comprised of 1437 patterns.

To solve the donor–recipient matching problem, the dependent variable is the class label which is equal to 1 when representing graft loss up to the first 15 days after the transplant, equal to 2 if the loss occurs between 15 days and 3 months, equal to 3 when the loss is after 3 months and before a year, and, finally, the last class corresponds to the patterns which do not present graft loss after the first year and is represented by label 4. The variables selected for the dataset can be seen in Table 1.

The choice of class limits for the dataset were not arbitrary (15 days, 3 months and a year); in addition to being considered as the most pertinent, Fig. 4 shows that the cumulative frequency slope of the graft loss curve changes strongly somewhere around those class limits. An important point is the limit located at 15 days since it is defined by experts as a critical point for survival or loss.

By analysing Fig. 4, it can be seen that the application of a regression-based technique is not suitable for the problem, due to the high number of points belonging to the *more than 1 year* category, which do not incorporate any knowledge about the real value of the number of days until either graft loss or death.

Table 1
Principal characteristics of the dataset: features considered, number of patterns and classes, etc. (number of patterns: 1437; number of classes: 4; number of features: 38; class distribution: {76, 76, 62, 1223}).

Attribute name	Type	Value
Recipient		
Age	Numeric	[18, 76]
Gender	Binary	0 = male; 1 = female
Body mass index	Numeric	[14, 68.3]
Diabetes mellitus	Binary	0 = absence; 1 = presence
Arterial hypertension	Binary	0 = absence; 1 = presence
Dialysis at transplant	Binary	0 = absence; 1 = presence
Etiology	Nominal	0 = virus C cirrhosis; 1 = alcohol; 2 = virus B cirrhosis; 3 = fulminant hepatic failure; 4 = primary biliary cirrhosis; 5 = primary sclerosing cholangitis; 6 = others
Portal thrombosis	Ordinal	0 = no; 1 = partial; 2 = complete
Waiting list time	Numeric	[0, 1978]
MELD (inclusion)	Numeric	[1, 46]
MELD (at transplant)	Numeric	[5, 50]
TIPS at transplant	Binary	0 = absence; 1 = presence
Hepatorrenal syndrome	Binary	0 = absence; 1 = presence
Upper abdominal surgery	Binary	0 = absence; 1 = presence
Pretransplant status performance	Nominal	0 = at home; 1 = hospitalized; 2 = hospitalized in ICU; 3 = hospitalized in ICU with mechanical ventilation
Cytomegalovirus	Binary	0 = absence; 1 = presence
Donor		
Age	Numeric	[10, 86]
Gender	Binary	0 = male; 1 = female
Body mass index	Numeric	[14.38, 53.35]
Diabetes mellitus	Binary	0 = absence; 1 = presence
Arterial hypertension	Binary	0 = absence; 1 = presence
Cause of exitus	Nominal	0 = brain trauma; 1 = cerebral vascular accident; 2 = anoxia; 3 = deceased vascular after cardiac death; 4 = others
Hospitalization length in ICU	Numeric	[0, 58]
Hypotension episodes	Binary	0 = absence; 1 = presence
High inotropic drug use	Binary	0 = absence; 1 = presence
Creatinine plasma level	Numeric	[0.1, 9.5]
Sodium plasma level	Numeric	[98, 187]
Aspartate transaminase level	Numeric	[1, 1090]
Alanine aminotransferase plasma level	Numeric	[2, 1400]
Total bilirubin	Numeric	[0.06, 4.2]
Hepatitis B	Binary	0 = absence; 1 = presence
Hepatitis C	Binary	0 = absence; 1 = presence
Cytomegalovirus	Binary	0 = absence; 1 = presence
Operative factors		
Multi-organ harvesting	Binary	0 = no; 1 = yes
Combined transplant	Binary	0 = no; 1 = yes
Complete or partial graft	Binary	0 = no; 1 = yes
Cold ischemia time	Ordinal	0 = <6 h; 1 = 6–12 h; 2 = >12 h
AB0 compatible transplant	Binary	0 = no; 1 = yes

The end-point variable is the time leading up to liver failure: (1) Before 15 days, (2) Between 15 days and 3 months, (3) Between 3 months and a year and (4) No graft failure presented after the first year.

All nominal and ordinal variables are transformed into binary ones.

3.2. Methods compared

The methods developed based on the cascade utility model using the SVM paradigm are the following:

- H-CascadeSVM: Hierarchical approach for the cascade utility model using SVM.
- ECOC-CascadeSVM: ECOC approach for the cascade utility model using SVM.

The results obtained have been compared with some of the most frequently used state-of-the-art nominal and ordinal methods, such as:

- Support Vector Classification with one-vs-one methodology (SVM(1v1)) and one-vs-all formulation (SVM(1vA)) [24]. These are the two main approaches for dealing with multiclass problems when using binary classifiers, and are also based on decomposing the original classification problem.
- Support Vector formulations for Ordinal Regression [16] (applying both implicit constraints (SVORIM) and explicit ones (SVOREX) depending on how the slack-variables are considered). These methodologies are commonly used in the ordinal regression paradigm, showing good generalization ability when considering ordinal measures. Their main idea is the computation of $K-1$ parallel discriminant hyperplanes and a set of ordered thresholds in order to separate the data.
- The Extended Binary Classification framework (EBCSVM) [19]: Reduction methodology that solves a set of binary problems by a single model by defining an extended binary dataset.
- The Extreme Learning Machine for Ordinal Regression (ELMOR) [29]. The Extreme Learning Machine paradigm has also been

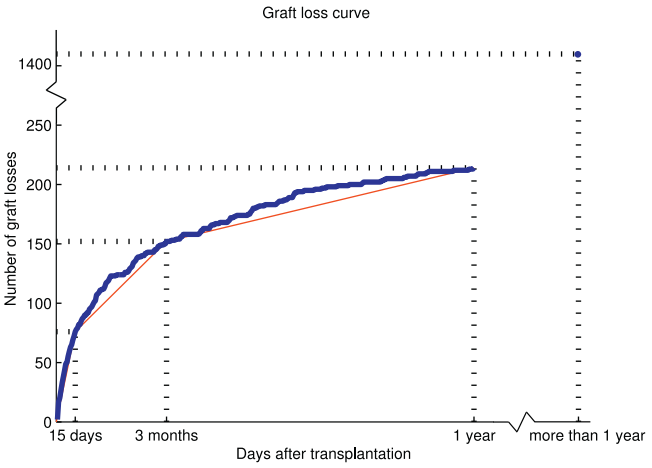


Fig. 4. Graphic showing the cumulative frequency of graft loss.

reformulated to deal with ordinal information by transforming the outputs of the model.

- The Proportional Odds Model (POM) [13]. This was the first threshold method applied to ordinal regression problems and it is based on a linear projection jointly trained with a set of thresholds by using a technique similar to that considered for nominal logistic regression.

The CascadeSVM proposals were implemented using Matlab, as well as the POM model available through the `mnrfit` function. The authors of SVORIM and SVOREX provide publicly available software,[1] as well as those by REDSVM.[2] The well-know `libsvm` implementation[3] was considered for all the different versions of SVM. The Matlab code for ELM[4] was adapted to implement ELMOR.

3.3. Evaluated measures

Several measures can be considered for evaluating ordinal classifiers. The most common ones in machine learning are the Mean Absolute Error (*MAE*) and the Mean Zero-one Error (*MZE*) [16,14,30,31], being $MZE = 1 - Acc$, where *Acc* is the accuracy or correct classification rate. However, as previously mentioned, these measures may not be the best option, for example, when measuring performance in the presence of class imbalances [32] and/or when the costs of different errors vary markedly. Because of this, the work makes use of measures of a different nature to evaluate classifier performance: accuracy (*Acc*) which measures overall performance, the geometric mean of the sensitivities (*GMS*) which is specially designed for unbalanced classification, the average mean absolute error (*AMAE*) which is a version of the ordinal *MAE* measure for unbalanced classification and, finally, Kendall's τ_b which is a correlation measure.

The metric used can be defined as follows:

- *Acc*: The correct classification rate or accuracy is the percentage of correctly classified patterns:

$$Acc = \frac{1}{N}\sum_{i=1}^{N}(I(y_i^* = y_i)),$$

where $I(\cdot)$ is the zero-one loss function, y_i is the desired output for pattern i, y_i^* is the prediction of the model and N is the total number of patterns in the dataset. *Acc* values vary from 0 to 1 and it represents global performance in the classification task. Apart from not taking into account category order, it has many disadvantages, especially where unbalanced problems are considered.

- *GMS*: The geometric mean of the sensitivities of each class is an average of the percentage of the correct classification of each of the classes:

$$GMS = \sqrt[K]{\prod_{k=1}^{K} S_k},$$

where $S_k = (1/N_k)\sum_{i=1}^{N_k}(I(y_i^* = y_i))$ is the sensitivity of the kth class, i.e. the percentage of patterns correctly predicted as belonging to the kth class with respect to the total number of examples in this class. This metric is of vital importance for evaluating the classifiers performance due to the highly imbalanced nature of the classification problem. In this case, the *GMS* will provide us with valuable information about which methods should be considered as trivial classifiers (i.e. a value of 0 for this metric indicates that the model is totally obviating at least one of the classes of the problem).

- *AMAE*: The average mean absolute error is the mean of *MAE* classification errors throughout the classes where *MAE* is the average absolute deviation of the predicted class from the true class (i.e. average absolute deviation in number of categories on the ordinal scale). *AMAE* was proposed by Baccianella et al. [32] to mitigate the effect of unbalanced class distributions. Let MAE_k be the *MAE* for a given kth class:

$$MAE_k = \frac{1}{N_k}\sum_{i=1}^{N_k}|\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|, \quad 1 \le k \le K,$$

where $\mathcal{O}(\mathcal{C}_k) = k$, $1 \le k \le K$, i.e. $\mathcal{O}(y_i)$ is the order of class label y_i. Then, the *AMAE* measure can be defined in the following way:

$$AMAE = \frac{1}{K}\sum_{k=1}^{K} MAE_k,$$

MAE values range from 0 to $K-1$, as do those of *AMAE*.

- τ_b: The Kendall's τ is a statistic used to measure the association between two measured quantities. Specifically, it is a measure of rank correlation [33]:

$$\tau_b = \frac{\sum_{i,j=1}^{n} c_{ij}^* c_{ij}}{\sqrt{\left(\sum_{i,j=1}^{n} c_{ij}^{*2}\right)\left(\sum_{i,j=1}^{n} c_{ij}^2\right)}},$$

where c_{ij}^* is +1 if $\mathcal{O}(y_i^*) > \mathcal{O}(y_j^*)$, 0 if $\mathcal{O}(y_i^*) = \mathcal{O}(y_j^*)$, and −1 if $\mathcal{O}(y_i^*) < \mathcal{O}(y_j^*)$ for $i, j = 1, \ldots, n$, and similar for c_{ij}. τ_b values ranging from −1 (maximum disagreement between the prediction and the true label), to 0 (no correlation between them) and to 1 (maximum agreement).

3.4. Evaluation and model selection

For the evaluation of the results, a 4-fold technique (stratified by the class and liver transplantation unit) to divide the data has been applied 10 times, using 75% of the patterns for training the

[1] http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm.
[2] http://home.caltech.edu/~htlin/program/libsvm/.
[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[4] http://www.ntu.edu.sg/home/egbhuang/.

model, and the remaining 25% for testing it. Hence, the results are taken as the mean and standard deviation of the measures over the 40 test sets.

The parameters of each algorithm are chosen using a 5-fold nested validation with each of the 40 training sets. The final parameter combination chosen is the one which obtains in mean the best average performance for the validation sets, where the metric used is the geometric mean of the sensitivities per class (*GMS*). The kernel selected for all the kernel methods is the Gaussian one, $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2)$ where σ is the kernel width. For every tested kernel method the kernel width was selected within these values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, as was the cost parameter associated to the SVM methods. For the ELMOR methodology, the sigmoidal base unit is used, and the number of hidden networks within the values $H \in \{5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.

4. Experimental results

Finally, the experiments are presented and discussed in this section, showing the results obtained and performing statistical tests in order to determine the statistical significance of the differences observed in the methods used.

The results of all the methods are included in Table 2. From these results, some conclusions can be drawn, but in order to do so, the unbalanced nature of the dataset must be taken into account along with the need to correctly classify minority classes. In this sense, recent results [34] have highlighted the problems of *Acc* and *MAE* when dealing with unbalanced and ordinal data.

4.1. Discussion

As stated above, several conclusions can be drawn from the results obtained (Table 2). First, one should take into account that several of the methodologies tested can be considered as trivial classifiers (for example the ELMOR method), since all the patterns are predicted to be in class 4. This can be seen by analysing the *AMAE* measure; specifically a mean *AMAE* of 1.5 means that all the patterns are classified in one of the extreme classes, being then in this case, $MAE_1 = 3$, $MAE_2 = 2$, $MAE_3 = 1$ and $MAE_4 = 0$ (where MAE_k is the *MAE* measure for class k), thus being the value of the $AMAE = (3+2+1+0)/4 = 1.5$. The fact that they are all classified in class 4 and not in class 1 is supported by the *Acc* value (85.11%), which is the one achieved by classifying only those patterns from class 4. In this situation, also the worst *GMS* possible is obtained: a value of 0.00%, which means that at least one class is totally misclassified for all the test sets. The SVM(1V1) and SVM(1VA) methods can also be considered trivial classifiers, since their mean *AMAE* and *Acc* are very close to the ones previously mentioned, as well as the POM method, which obtained even worse values for *Acc* and *AMAE* and a negative value of τ_b, which means that the predictions and true labels are negatively correlated.

On the other hand, it can be seen that the use of techniques specially designed for ordinal regression helped to improve the performance (in terms of *GMS*, *AMAE* and τ_b) of some state-of-the-art nominal methods like SVM(1V1) and SVM(1VA). In particular, the decrease that has been seen in the *AMAE* measure means that fewer serious ordinal errors are committed. Furthermore, the computation of a more complex model by combining some simpler ones (like the proposal used in this paper or the REDSVM methodology) seemed to help to improve the performance of the ordinal algorithms, due to the great complexity and unbalanced nature of the dataset.

Indeed, the application of the ECOC framework in the cascade utility model with the hinge-loss function has proved to work better than the hierarchical model in this specific application (in terms of *GMS*, *AMAE* and τ_b). As mentioned previously, this can be attributed to the fact that the hierarchical model would be ideally biased towards the 4th class which is actually the majority one, achieving then better mean accuracy but worse values in the remaining metrics; it can also be due to the choice of the hinge-loss function, which is more appropriate for SVM models.

In order to choose the method that performs best for this application, one should take into consideration that obviously, although ELMOR, SVM(1V1), SVM(1VA) and POM obtained the best results in terms of accuracy, these algorithms should not be considered since the final classification is very biased towards class 4. Furthermore, the *Acc* measure do not contribute much information in this situation, although it is still necessary for the application to achieve an acceptable accuracy value. Due to these reasons and on analysing the results, the ECOC-CascadeSVM is considered to be the one with the best performance, and the best model obtained (that can also be seen in Table 2) is chosen for the construction of the organ allocation system.

Nonetheless, the results obtained (for example the low τ_b values achieved, even for the best model) indicate: the true complexity of the problem dealt within this study; also the fact that other characteristics that have not been included in the dataset may be of importance for the characterization of the dependent variable (survival time after transplantation).

Finally, by comparing the results with those obtained in a preliminary study using ordinal regression [35], it can be seen that the use of the combined information from Spanish and English hospitals help to obtain better and more robust models than those obtained only from Spanish hospitals, demonstrating then that the information has been properly combined and that a transnational model could be more robust and, therefore, could better generalize on unseen data.

4.2. Statistical tests

In order to determine the statistical significance of the differences observed in the methods used, statistical tests have been performed for each metric selected (*Acc*, *GMS*, *AMAE* and τ_b). First of all, there has been an analysis to determine whether each of the different performance metrics selected for all the methods followed a normal distribution. In none of these cases can a normal distribution be assumed by using a Kolmogorov–Smirnov's test (KS-test) at a significance level $\alpha = 0.05$. As a consequence, a non-parametric Friedman's test for dependent samples was selected in order to check if the method applied does significantly affect the results obtained. The test concludes that these differences in ranking are significant (with a p-value = 0.00). Hence, the statistical analysis ends applying the Wilcoxon's signed-rank test for all pairs of algorithms. For this test, the significance level was adjusted to control the family-wise error: α was divided by the number of comparisons made minus one,

$$\alpha_{Wilcoxon} = \frac{\alpha}{\frac{nAlgs*(nAlgs-1)}{2} - 1},$$

where *nAlgs* is the number of algorithms used [36]. The results obtained can be seen in Table 3. For each algorithm these results include the number of algorithms statistically outperformed (Wins, W), the number of draws (non-significant differences, D) and the number of losses (number of algorithms that outperform the method, L).

On analysing the results, it can be seen that the methods obtaining the best performance in *Acc* can not be considered acceptable solutions for the problem, due to the triviality of the models obtained; consequently, they do not present wins (in terms of statistical differences) in any of the other metrics selected. In

Table 2
Means and standard deviations ($Mean_{SD}$) for the different methods selected in the test sets.

Method	*Acc*(%)	*GMS*(%)	*AMAE*	τ_b
Nominal methods				
SVM(1V1)	$84.82_{1.32}$	$0.00_{0.00}$	$1.497_{0.017}$	$0.001_{0.014}$
SVM(1VA)	$84.43_{1.95}$	$0.00_{0.00}$	$1.494_{0.022}$	$0.004_{0.028}$
Ordinal methods				
SVORIM	$79.99_{8.51}$	$1.03_{4.64}$	$1.445_{0.096}$	$0.018_{0.036}$
SVOREX	$79.99_{8.52}$	$1.03_{4.64}$	$1.445_{0.096}$	$0.019_{0.037}$
REDSVM	$79.93_{8.62}$	$1.07_{4.82}$	$1.445_{0.096}$	$0.017_{0.035}$
POM	$83.35_{0.63}$	$0.00_{0.00}$	$1.504_{0.025}$	$-0.032_{0.030}$
ELMOR	$\mathbf{85.11}_{0.24}$	$0.00_{0.00}$	$1.500_{0.000}$	$0.000_{0.000}$
H-CascadeSVM	$78.30_{5.57}$	$3.80_{6.36}$	$1.435_{0.058}$	$0.033_{0.043}$
ECOC-CascadeSVM	$64.18_{2.94}$	$\mathbf{7.46}_{9.47}$	$\mathbf{1.345}_{0.083}$	$\mathbf{0.037}_{0.043}$
Best model				
ECOC-CascadeSVM	65.46	24.69	1.154	0.092

The best result is in **bold** face and the second best result is in *italics*.

Table 3
Number of wins (W), draws (D) and losses (L) when comparing the different methods using the Wilcoxon's signed-rank test with $\alpha = 0.05$.

	Wilcoxon's signed-rank test (W/D/L)			
	Acc	*GMS*	*AMAE*	τ_b
SVM(1V1)	**6/2/0**	0/6/2	0/3/5	1/5/2
SVM(1VA)	*3/5/0*	0/6/2	0/6/2	1/5/2
SVORIM	1/5/2	*0/8/0*	3/4/1	*2/6/0*
SVOREX	1/5/2	*0/8/0*	3/4/1	*2/6/0*
REDSVM	1/5/2	*0/8/0*	3/4/1	*2/6/0*
POM	2/3/3	0/6/2	0/3/5	0/0/8
ELMOR	**6/2/0**	0/6/2	0/3/5	1/2/5
H-CascadeSVM	1/3/4	**4/4/0**	*4/3/1*	**4/4/0**
ECOC-CascadeSVM	0/0/8	**4/4/0**	**8/0/0**	**4/4/0**

The best result is in **bold** face and the second best result in *italics*.

general, the methods which stand out as being the most competitive approaches are the ECOC-CascadeSVM and the H-CascadeSVM methods. The first of these methods obtains the best performance in 3 out of the 4 metrics analysed and the second one obtains the best performance in 2 metrics and a second best result. Comparing these two methods, it can be observed that the hierarchical model obtains better results in *Acc* than the ECOC version, but presents a dismal performance in *AMAE* compared to the ECOC version; this may be due to the unbalanced nature of the dataset. For all these reasons, in our opinion, the ECOC-CascadeSVM method is the best solution for the problem, since it presents an acceptable balance in all metrics, thus showing competitiveness in all aspects due to the different nature of the metrics selected.

5. Proposed system for organ allocation

D–R matching occurs at the time of organ procurement. However, since the MELD score obviates donor characteristics, the assignment of a donor to the patient listed first on the list of the most seriously ill can not be considered true D–R matching. Therefore, in a MELD-based allocation policy, a concrete D–R combination does not necessarily provide the best combination in terms of outcome. Based on the best model obtained in the present study, a novel liver allocation system is proposed. The first stage of the system is the selection of the first k recipients on the waiting list (in our case, we consider the case of $k = 5$). Note that patients on the waiting list are sorted according to the MELD score, and in case of draws by considering the length of time spent on the waiting list. After this, these k recipients are evaluated with the best model obtained in order to predict graft survival after transplantation. The allocation is performed by choosing the recipient with the highest predicted class (i.e. the patient that presents the highest predicted time leading up to graft failure). In case of draws between two or more recipients, the one with a higher MELD value is selected. This new system is intended to complement the assignment of the MELD score and to take donor and operative factors into account. Fig. 5 underlines the general ideas of the proposed liver allocation system.

5.1. Example of how the system works

In this subsection, the proposed system is confronted with five different situations in order to analyse its response and behaviour. These situations are made up of five randomly selected recipients chosen from the dataset with MELD values between 9 and 35 points. The characteristics of the recipients are presented in Table 4 (recipients 1 to 25). More specifically, the following situations were tested: recipients with MELD value lower than 20, recipients with MELD values between 20 and 23, recipients with values between 24 and 26, recipients with values between 28 and 35 and recipients with the same MELD value (in this case, a value of 27). The responses of the recipients in these situations were tested with ten potential donors with non-extended (1–5) and extended criteria (6–10), randomly selected from the dataset. Note that extended criteria donors are those that present at least two of the following restrictions: Age >75 years; hospitalization length in ICU > 4 days; high inotropic drug use = 1; BMI > 30; Cold Ischemia Time = 2 (>12 h). The characteristics of the selected donors are shown in Table 5.

The result of simulating the system in the five previously defined situations can be seen in Table 6, where graft survival after transplantation is predicted for each donor–recipient pair. Several conclusions can also be drawn from this experiment. Concerning

Information needed for the allocation

1: Characteristics of patients in waiting list
2: Characteristics of donor with organ to be allocated
3: Best model obtained from previous training

Proposed allocation process

1. Initialize the number of patients to consider in the system (k)
2. Sorted_list: Sort Waiting_list by MELD and waiting list time
3. Possible_recipients: Select first k recipients of Sorted_list
4. Predictions: Evaluate Possible_recipients with Best_model and Donor_characteristics
5. Best_recipients: Recipients with highest survival time predicted
6. MELD_values: Compute MELD for selected Best_recipients
7. Selected_recipient: Select recipient of Best_recipients with greater MELD
8. Assign organ to Selected_recipient

Fig. 5. Pseudocode of the proposed liver allocation system.

Table 4
Recipients selected from the dataset used to exemplify the allocation system (recipient characteristics).

	A	G	BMI	DM	AH	D	E	PT	WT	M	MO	TT	HS	UAS	SP	C
Rec1	66	0	29.86	0	1	0	1	0	133	19	19	0	1	0	0	1
Rec2	64	0	26.49	0	0	0	0	0	533	9	18	1	0	0	1	1
Rec3	58	0	29	0	0	0	2	1	159	11	14	0	0	1	0	1
Rec4	67	1	38.10	0	0	0	1	0	106	7	9	0	1	0	0	1
Rec5	65	1	23	0	0	0	5	1	23	9	9	0	0	1	0	1
Rec6	47	0	30.86	0	0	0	5	0	243	9	23	0	0	0	0	1
Rec7	19	1	29.21	0	1	0	5	0	95	23	23	0	0	0	0	0
Rec8	66	0	25.68	0	0	0	0	1	14	22	22	0	0	0	0	1
Rec9	51	1	25.34	0	0	0	0	0	333	14	21	0	0	0	1	1
Rec10	56	0	33.41	0	0	0	5	0	2	21	20	0	0	1	0	1
Rec11	32	0	29.21	0	0	0	3	0	1	27	27	0	0	0	3	0
Rec12	57	1	40.05	0	1	1	6	0	768	24	26	0	0	0	0	1
Rec13	58	0	28.73	0	0	0	5	0	43	25	25	0	0	0	0	1
Rec14	63	1	34.60	0	1	0	3	0	2	26	25	0	0	1	2	1
Rec15	61	0	25.43	0	0	0	0	0	121	24	24	0	0	0	0	1
Rec16	28	0	28.90	0	0	0	3	0	2	33	35	0	0	0	3	1
Rec17	58	1	29.13	1	0	1	6	0	2	34	34	0	1	1	2	1
Rec18	46	1	35.43	0	1	0	3	0	1	32	32	0	0	0	3	1
Rec19	50	0	41	0	0	0	1	0	3	29	29	0	0	0	1	1
Rec20	36	0	30.86	0	1	0	6	1	1	28	28	0	0	0	3	1
Rec21	51	0	30.52	0	1	0	1	0	73	27	27	0	0	0	1	1
Rec22	44	0	35.43	0	0	0	1	0	63	13	27	0	1	0	0	0
Rec23	51	0	26	0	0	0	1	0	32	24	27	0	0	0	0	0
Rec24	57	1	44.44	0	0	0	4	0	10	28	27	0	1	0	0	0
Rec25	52	0	41.62	0	0	0	0	0	7	26	27	0	1	0	1	1

A: age; G: gender; BMI: body mass index; DM: diabetes mellitus; AH: arterial hypertension; D: dialysis at transplant; E: etiology; PT: portal thrombosis; WT: waiting list time; M: MELD score at listing; MO: MELD score at the operation; TT: TIPS at transplant; HS: hepatorrenal syndrome; UAS: upper abdominal surgery; SP: status performance pretransplant; C: cytomegalovirus.

Table 5
Donors and surgery factors selected from the dataset used to exemplify the allocation system (donor and operation characteristics).

	A	G	BMI	DM	AH	CE	ICU	Hy	In	Cr	NA	AST	ALT	TB	AHB	HCV	C	MU	CT	CP	CIT	AB0
Don1	39	0	26.42	0	0	1	0	0	1	0.9	141	30	45	1	1	0	0	0	0	1	0	1
Don2	73	1	29.61	0	1	1	1	0	0	1.2	136	29	29	1.2	1	0	1	1	0	0	1	1
Don3	46	0	24.05	0	0	4	0	0	0	0.73	142	19	20	0.31	0	0	1	1	0	0	0	1
Don4	55	1	21.08	0	0	1	1	0	0	0.7	139	28	21	1.6	0	0	1	0	0	0	1	1
Don5	25	0	21.60	0	0	0	1	1	0	1	148	112	83	0.9	0	0	0	1	0	0	1	1
Don6	78	1	41.11	0	0	1	8	0	0	0.85	159	42	30	0.3	0	0	1	1	0	0	1	1
Don7	75	0	31.14	0	0	0	8	0	0	0.7	139	29	36	1	0	0	1	0	0	0	1	1
Don8	79	0	28.08	0	1	1	3	1	1	0.8	152	25	27	1	0	0	1	1	0	0	1	1
Don9	35	0	31.14	0	0	0	6	0	1	0.8	150	105	80	0.7	1	0	1	1	0	0	2	1
Don10	67	0	31.14	0	1	1	30	0	1	1.1	158	363	300	2.6	1	0	1	1	0	0	0	1

A: age; G: gender; BMI: body mass index; DM: diabetes mellitus; AH: arterial hypertension; CE: cause of exitus; ICU: hospitalisation length in intensive care unit; Hy: hypotension episodes >1 h <60 mmHg; In: high inotropic drug use; Cr: creatinine plasma level; NA: sodium plasma level; AST: aspartate transaminase level; ALT: alanin aminotransferase plasma level; TB: total bilirubin; AHB: hepatitis B (core Ab positive); HCV: hepatitis C (positive serology); C: cytomegalovirus; MU: multi-organ harvesting; CT: combined transplant; CP: complete or partial graft; CIT: cold ischemia time; AB0: AB0 compatible transplant.

donors, it can be seen that there are clearly some donor characteristics that favour graft survival after transplantation, independently in most of the cases of recipient characteristics (the case of Don1 and Don5). It is remarkable in this case that these donors do not present extended criteria (i.e. donors that do not present the combination of several marginal factors that can result in graft loss), and the fact that there seems to be some incompatibilities with some recipients (see for example Don5, that almost always shows good performance with any recipient, except for Rec4, Rec5 and Rec17). In contrast, Don7 and Don10, that are extended criteria donors, present poor predicted graft survival in general for any recipient. Analysing these donors, they are seen to be of considerably advanced age, and have experienced long hospitalisation stays in the intensive care unit. These data are consistent with results reported in literature where age is an important factor contributing to the donor risk index. Similarly, prolonged ICU hospitalization is a strong predictor of early graft dysfunction and poor initial functioning that increased post-transplant hospital costs.

With regard to recipients, it can also be seen that some patients appear to be compatible with almost all donors (this is the case of Rec12). Nevertheless, some incompatibilities can be found when performing the matching (Rec12 and Don10), which simply indicates the need to take into account both donor and recipient characteristics. These mismatches can not be detected under a MELD policy because it only includes recipient characteristics and the donor–recipient pairs are not well-categorized in accordance with the net benefit of their combinations. MELD was not designed for D–R matching and therefore it is a suboptimal tool for this task. In addition, recipients presenting a high MELD value are compatible with different donors too (see Rec16, who presents the highest MELD and is compatible with Don1, Don2, Don3, Don5, Don6 and Don8). This is congruent with the current survival benefit allocation system, which considers both waiting list mortality (urgency principle) and post-transplant mortality (utility principle). The survival benefit (SB) computes the difference between the mean lifetime with and without a liver transplantation (LT). This new allocation system seeks to minimize futile LT, giving primary attention to patients with the best predicted lifetime gained due to transplantation. Under a SB model, an allocated graft goes to the patient with the greatest difference between the predicted post-transplant lifetime and the predicted waiting list lifetime for this specific donor. A conclusion about the survival benefit is that higher-MELD patients

Table 6
Simulation examples of the proposed system, where different situations are considered. The predicted values of the system are reported, where 1 corresponds to the "less than 15 days survival" class, 2 to the "between 15 days and 3 months", 3 to the "between 3 months and 1 year", and finally, 4 represents the "over 1 year survival" group.

Rec(MELD)	Don1	Don2	Don3	Don4	Don5	Don6	Don7	Don8	Don9	Don10
Situation 1: recipients with MELD values lower than 20										
Rec1(19)	4	2	4	4	4	2	2	4	3	2
Rec2(18)	4	3	4	4	4	3	2	2	4	2
Rec3(14)	4	4	1	4	4	4	2	4	4	4
Rec4(9)	4	4	4	4	2	2	2	4	3	2
Rec5(9)	4	2	2	2	2	2	2	4	1	2
Allocation	Rec1	Rec3	Rec1	Rec1	Rec1	Rec3	Rec1	Rec1	Rec2	Rec3
Situation 2: recipients with MELD values between 20 and 23										
Rec6(23)	4	3	4	3	4	4	3	2	3	2
Rec7(23)	4	3	4	3	4	4	3	3	4	3
Rec8(22)	4	4	1	4	4	4	2	4	1	4
Rec9(21)	4	2	1	2	4	4	2	4	4	4
Rec10(20)	4	2	1	2	4	2	2	2	3	2
Allocation	Rec6	Rec8	Rec6	Rec8	Rec6	Rec6	Rec6	Rec8	Rec7	Rec8
Situation 3: recipients with MELD values between 24 and 27										
Rec11(27)	4	3	4	3	4	4	3	2	3	2
Rec12(26)	4	4	4	4	4	4	4	4	4	2
Rec13(25)	4	4	1	4	4	4	2	4	1	4
Rec14(25)	4	2	1	2	4	4	2	4	4	4
Rec15(24)	4	2	1	2	4	2	2	2	3	2
Allocation	Rec11	Rec12	Rec11	Rec12	Rec11	Rec11	Rec12	Rec12	Rec12	Rec13
Situation 4: recipients with MELD values between 28 and 35										
Rec16(35)	4	4	4	1	4	4	1	4	1	2
Rec17(34)	4	2	2	2	2	4	2	2	2	4
Rec18(32)	4	4	4	4	4	4	2	4	4	2
Rec19(29)	4	1	1	1	4	1	1	1	1	2
Rec20(28)	4	4	4	4	4	4	1	4	4	2
Allocation	Rec16	Rec16	Rec 16	Rec18	Rec16	Rec16	Rec17	Rec16	Rec18	Rec17
Situation 5: recipients with the same MELD values (MELD value = 27)										
Rec21(27)	4	1	4	1	4	1	1	1	4	2
Rec22(27)	4	3	4	3	4	3	3	3	3	3
Rec23(27)	4	2	4	2	4	4	2	2	4	2
Rec24(27)	4	4	4	4	4	2	3	4	3	3
Rec25(27)	4	1	1	4	4	2	2	2	2	4
Allocation	Rec21	Rec24	Rec21	Rec24	Rec21	Rec23	Rec22	Rec24	Rec21	Rec25

have a significant SB from transplantation regardless of the donor risk index; also lower-MELD candidates who receive higher-DRI organs experience higher mortality and do not demonstrate significant SB.

As a final remark, it can be seen that the predicted outputs of this model are diverse for the most part, and thus could be useful for breaking a deadlock in case of draws (as in situation 5). Furthermore, although it has been seen that considering the characteristics of donors and recipients independently can be useful for predicting graft survival (since it has been noticed that there are some determining factors in these situations), the use of both sources of information could be more useful and beneficial for the survival principle.

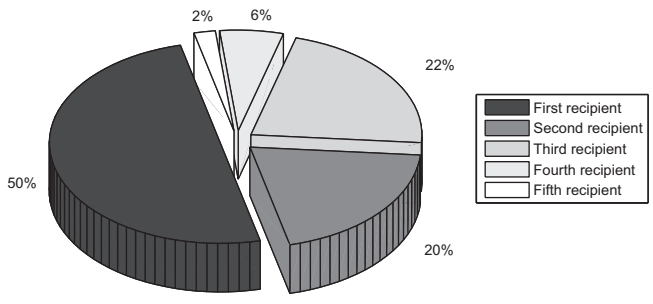


Fig. 6. Pie chart of the assignment percentage corresponding to each recipient.

Fig. 6 shows the assignment percentage corresponding to each recipient. This figure shows that the new system proposed works well according to the MELD score in 50% of the cases. However, the remaining recipients with lower MELD scores have a better chance than the most urgent candidates with some donors. The system proposed ensures that no recipient will remain indefinitely on the waiting list. In other words, this new system essentially considers the urgent principle (MELD), but gives opportunities to healthier recipients with better potential outcomes (utility or benefit principle).

6. Conclusions

Ordinal regression analysis and the Support Vector Machine paradigm have been used as machine learning techniques for predicting graft survival after liver transplantation taking into account donors and recipients characteristics and other operative factors concerning the transplant, through the construction of a dataset compound of donor–recipient pairs from Spanish and UK hospitals. More specifically, the classification model has been designed to deal with imbalanced and ordinal data to provide a fairer decision maker when allocating an organ to a recipient and the evaluation of

the different classifiers has been accomplished by considering a set of four metrics designed for imbalanced and ordinal classification problems to avoid trivial solutions focused on improving overall error. The best model obtained from the whole set of methodologies tested was used in conjunction with the MELD score, which is the cornerstone of the current allocation policy globally. The experiments show that, although it is a really complex problem which may need more information in order to perform perfectly, the proposal is able to generalize well on unseen data, helps to avoid draws caused by the MELD score and does seem to work well in more realistic situations. The final rule-based system, which as said uses the MELD score and the best performing machine learning model, will consider the allocation of the organ to one of the first recipients in waiting list (these patients being ranked using the MELD score to estimate the patients severity) and the decision is made selecting the patient that presents a higher survival probability (note that we considered 4 different levels of time survival after transplantation).

As future work, a sensitivity analysis of the best model can be developed in order to determine the most important variables for the end-point variable. Moreover, this study can be extended by considering other liver transplantation centres in the European Union to unify the procedure and create a more generic supranational organ allocation system. Finally, a different source of information concerning the transplant could be also used (post-transplant information) by reformulating the proposed algorithm to use the so-called privileged sources of information (note that this information can not be used directly in the model because it will not be available when deciding the matching).

Acknowledgments

This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain). Manuel Cruz-Ramírez's research has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science), grant reference AP2009-0487. The authors M. Pérez-Ortiz, M. Cruz-Ramírez and M.D. Ayllón-Terán have contributed equally to the preparation of this paper.

References

[1] S.B. Kotsiantis, I.D. Zaharakis, P.E. Pintelas, Machine learning: a review of classification and combining techniques, Artificial Intelligence Review 26 (3) (2006) 159–190.
[2] A. Yardimci, Soft computing in medicine, Applied Soft Computing 9 (3) (2009) 1029–1043.
[3] M.-H. Tseng, H.-C. Liao, The genetic algorithm for breast tumor diagnosis – the case of DNA viruses, Applied Soft Computing 9 (2) (2009) 703–710.
[4] C.-J. Su, C.-Y. Wu, Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring, Applied Soft Computing 11 (1) (2011) 315–325.
[5] S. Feng, N. Goodrich, J. Bragg-Gresham, D. Dykstra, J. Punch, M. DebRoy, S. Greenstein, R. Merion, Characteristics associated with liver graft failure: the concept of a donor risk index, American Journal of Transplantation 6 (4) (2006) 783–790.
[6] P. Kamath, W. Kim, The Model for End-stage Liver Disease (MELD), Hepatology 45 (3) (2007) 797–805.
[7] R.W. Busuttil, K. Tanaka, The utility of marginal donors in liver transplantation, Liver Transplant 9 (7) (2003) 651–663.
[8] J. Briceño, G. Solorzano, C. Pera, A proposal for scoring marginal liver grafts, Transplant International 13 (2000) S249–S252.
[9] A. Rana, M.A. Hardy, K.J. Halazun, D.C. Woodland, L.E. Ratner, B. Samstein, J.V. Guarrera, R.S. Brown, J.C. Emond, Survival outcomes following liver transplantation (SOFT) score: a novel method to predict patient survival following liver transplantation, American Journal of Transplantation 8 (12) (2008) 2537–2546.
[10] P. Dutkowski, C. Oberkofler, K. Slankamenac, M. Puhan, E. Schadde, B. Mllhaupt, A. Geier, P. Clavien, Are there better guidelines for allocation in liver transplantation? A novel score targeting justice and utility in the model for end-stage liver disease era, Annals of Surgery 254 (5) (2011) 745–753.
[11] M. Cruz-Ramírez, C. Hervás-Martínez, J. Fernandez-Caballero, J. Briceño, M. de la Mata, Multi-objective evolutionary algorithm for donor–recipient decision system in liver transplants, European Journal of Operational Research 222 (2) (2012) 317–327.
[12] D.E. Schaubel, M.K. Guidinger, S.W. Biggins, J.D. Kalbfleisch, E.A. Pomfret, P. Sharma, R.M. Merion, Survival benefit-based deceased-donor liver allocation, American Journal of Transplantation 9 (4 Pt 2) (2009) 970–981, http://dx.doi.org/10.1111/j.1600-6143.2009.02571.x.
[13] P. McCullagh, Regression models for ordinal data, Journal of the Royal Statistical Society, Series B (Methodological) 42 (2) (1980) 109–142.
[14] B.-Y. Sun, J. Li, D.D. Wu, X.-M. Zhang, W.-B. Li, Kernel discriminant learning for ordinal regression, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 906–910.
[15] P. McCullagh, J.A. Nelder, Generalized Linear Models, 2nd Edition, Monographs on Statistics and Applied Probability, Chapman & Hall/CRC, 1989.
[16] W. Chu, S.S. Keerthi, Support vector ordinal regression, Neural Computation 19 (3) (2007) 792–815.
[17] E. Frank, M. Hall, A simple approach to ordinal classification, in: Proceedings of the 12th European Conference on Machine Learning, EMCL'01, 2001, pp. 145–156.
[18] W. Waegeman, L. Boullart, An ensemble of weighted support vector machines for ordinal regression, International Journal of Computer Systems Science and Engineering 3 (1) (2009) 7–11.
[19] L. Li, H.T. Lin, Ordinal regression by extended binary classification, Advances in Neural Information Processing Systems 19 (2007) 865–872.
[20] J.S. Cardoso, J.F.P. da Costa, Learning to classify ordinal data: the data replication method, Journal of Machine Learning Research 8 (2007) 1393–1429.
[21] H.-T. Lin, L. Li, Reduction from cost-sensitive ordinal ranking to weighted binary classification, Neural Computation 24 (5) (2012) 1329–1367.
[22] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: D. Haussler (Ed.), Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, ACM Press, Pittsburgh, PA, 1992, pp. 144–152.
[23] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[24] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transaction on Neural Networks 13 (2) (2002) 415–425.
[25] H. Wu, H. Lu, S. Ma, A practical SVM-based algorithm for ordinal regression in image retrieval, in: Proceedings of the eleventh ACM international conference on Multimedia (Multimedia 2003), 2003, pp. 612–621.
[26] Y. Kwon, I. Han, K. Lee, Ordinal pairwise partitioning (opp) approach to neural networks training in bond rating, Intelligent Systems in Accounting Finance and Management 6 (1) (1997) 23–40.
[27] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1) (1995) 263–286.
[28] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2001) 113–141.
[29] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, X. Wang, Ordinal extreme learning machine, Neurocomputation 74 (1-3) (2010) 447–456.
[30] P.A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, C. Hervás-Martínez, An experimental study of different ordinal regression methods and measures, in: 7th International Conference on Hybrid Artificial Intelligence Systems (HAIS), 2012.
[31] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutiérrez, A preliminary study of ordinal metrics to guide a multi-objective evolutionary algorithm, in: 11th International Conference on Intelligent Systems Design and Applications (ISDA2011), 2011, pp. 1176–1181.
[32] S. Baccianella, A. Esuli, F. Sebastiani, Evaluation measures for ordinal regression, in: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications, ISDA'09, 2009, pp. 283–287.
[33] M.G. Kendall, Rank Correlation Methods, Hafner Press, New York, 1962.
[34] J.S. Cardoso, R. Sousa, Measuring the performance of ordinal classification, International Journal of Pattern Recognition and Artificial Intelligence 25 (8) (2011) 1173–1195.
[35] M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez, J. Briceño, M. de la Mata, An ensemble approach for ordinal threshold models applied to liver transplantation, in: Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN), 2012, pp. 2795–2802.
[36] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

3.4. Log-gamma distribution optimisation via maximum likelihood for ordered probability estimates

The following paper presents a post-learning procedure for threshold models, which are the most common case for ordinal classification. The reason why it is included in this chapter is that it could be used to improve the different decompositions presented in the previous sections, given that threshold methods are used as base methodologies. These models are based on the idea of projecting the patterns to a line, which is thereafter divided into intervals using a set of biases or thresholds. This paper proposes a general likelihood-based optimisation framework to better fit probability distributions for ordered categories. The motivation for the optimisation method proposed is to derive a general probability estimation framework for projected patterns obtained by the previously computed projection which can be used in conjunction with all threshold models (linear or nonlinear). Usually, for cumulative link models, the final response depends on the link function considered [1] (logit, probit, complementary log-log, negative log-log, cauchit functions, etc). However, the optimal choice of this distribution will depend on the distribution of the patterns itself, making their choice an arduous task. Thus, a more suitable and realistic option, which is barely explored in the literature, is to use a generalised link function and optimise it according to the data. To do so, in this paper a specific probability distribution (log-gamma [75]) is used, which includes a parameter that modifies the shape of the function and that generalises three commonly used link functions (log-log, probit and complementary log-log). Usually, the value of these parameters is the same for all classes [75]. However, we propose to consider this generalised cumulative distribution function for modelling probabilities of projected patterns, but allowing a different parameter for each class which will ideally provide better probability estimates for ordered categories.

The experiments performed in this paper show that the methodology is not only useful to provide a probabilistic output of the classifier but also to improve the performance of threshold models when reformulating the prediction rule to take these probabilities into account. More specifically, the results suggests that a per-class bias and probability distribution optimisation is indeed a crucial step for the base methodology, leading to an improvement of performance.

Finally, it should be said that this technique could be used in conjunction with any threshold method or with any probability-based decomposition method (as the ones considered in this paper). It is clear that if the probability estimation stage is improved, an ensemble method based on the union of several probabilities will notice this improvement as well, therefore, the combination of both is very interesting.

Log-Gamma Distribution Optimisation via Maximum Likelihood for Ordered Probability Estimates*

M. Pérez-Ortiz, P.A. Gutiérrez, and C. Hervás-Martínez

University of Córdoba, Dept. of Computer Science and Numerical Analysis, Rabanales Campus, Albert Einstein building, 14071 - Córdoba, Spain
{i82perom,pagutierrez,chervas}@uco.es

Abstract. Ordinal regression considers classification problems where there exist a natural ordering between the categories. In this learning setting, thresholds models are one of the most used and successful techniques. These models are based on the idea of projecting the patterns to a line, which is thereafter divided into intervals using a set of biases or thresholds. This paper proposes a general likelihood-based optimisation framework to better fit probability distributions for ordered categories. To do so, a specific probability distribution (log-gamma) is used, which generalises three commonly used link functions (log-log, probit and complementary log-log). The experiments show that the methodology is not only useful to provide a probabilistic output of the classifier but also to improve the performance of threshold models when reformulating the prediction rule to take these probabilities into account.

Keywords: Ordinal regression, discriminant learning, log-gamma, probability estimation, maximum likelihood.

1 Introduction

The classification of patterns into naturally ordered labels is referred to as ordinal regression or ordinal classification. This learning paradigm, although still mostly unexplored, is spreading rapidly and receiving a lot of attention from the pattern recognition and machine learning communities [1], given its applicability to real world problems. Thresholds models [1,2,4] are one of the most common methodologies for classification problems where the categories exhibit an ordering. The main assumption made by these methods is that an underlying real-valued outcome (also known as latent variable) exists for the ordered crisp categories, although it is unobservable. Consequently, these methodologies try to estimate two elements:

– A function $g(\mathbf{x})$ to predict the nature of the latent variable.

* This work has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain).

M. Polycarpou et al. (Eds.): HAIS 2014, LNAI 8480, pp. 454–465, 2014.

– A vector of thresholds $\mathbf{b} = (b_1, b_2, \ldots, b_{K-1}) \in \mathbb{R}^{K-1}$ (where K is the number of classes in the problem) to represent the intervals in the range of $g(\mathbf{x})$, where $b_1 \leq b_2 \leq \ldots \leq b_{K-1}$.

For example, if the categories of an ordinal regression problem of age estimation are {*young*, *adult*, *old*}, a threshold model would try to uncover the latent variable related to the actual age of the person and the thresholds would divide this latent variable into the considered categories.

There has been a great deal of work of probabilistic linear regression models for ordinal response variables [4,3]. These models are known as Cumulative Link Models (CLMs) and they are based on the idea of modelling the cumulative probability of each pattern to belong to a class lower than the class which is being considered. The proportional odds model (POM) [4] is the first model in this category, being a probabilistic model which leads to linear decision boundaries, given that the latent function $g(\cdot)$ is a linear model. This probabilistic model resembles the threshold model structure, although the linearity of $g(\cdot)$ can limit its applicability for real datasets. Other threshold models have been considered in the literature for ordinal regression, such as the support vector machine (SVM) reformulation [5] or the kernel discriminant learning one [2], which result into nonlinear decision boundaries based on a nonlinear latent function $g(\cdot)$. For these models, the thresholds are chosen so as to perform the classification task, without directly providing probability estimates for the patterns. In this sense, the motivation for the optimisation method proposed in this paper is to derive a general probability estimation framework for projected patterns obtained by $g(\cdot)$ which can be used in conjunction with all threshold models (linear or nonlinear).

In CLMs, the final response depends on the link function considered (logit, probit, complementary log-log, negative log-log or cauchit functions) [3]. The optimal choice of this distribution will directly depend on the distribution of the patterns itself. A more suitable option, which is barely explored in the literature, is a generalised link function (such as the one used in this paper: the log-gamma distribution [6], which generalises the probit, log-log and complementary log-log links). The log-gamma distribution depends on a parameter, q, which modifies the shape of the function. Usually, the value of q is the same for all classes [6]. We propose to consider this generalised cumulative distribution function for modelling probabilities of projected patterns, but allowing a different q for each class which will ideally provide better probability estimates for ordered categories.

The rest of the paper is organised as follows: Section 2 presents the methodology proposed, while Section 3 presents and discusses the experimental results. The last section summarises the main contributions of the paper.

2 Methodology

The goal in ordinal classification is to assign an input vector $\mathbf{x}$ to one of K discrete classes $\mathcal{C}_k, k \in \{1, \ldots, K\}$ where there exists a given ordering between the labels $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$, $\prec$ denoting this order information. Hence the

objective is to find a prediction rule $C : \mathcal{X} \rightarrow \mathcal{Y}$ by using an i.i.d. training sample $X = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ where N is the number of training patterns, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $\mathcal{X} \subset \mathbb{R}^k$ is the k-dimensional input space and $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$ is the label space. For convenience, denote by $\mathbf{X}_i$ to the set of patterns belonging to $\mathcal{C}_i$.

The optimisation methodology presented in this paper can be used with a wide range of ordinal regression models in order to obtain probability estimates, provided they resemble the threshold model structure. However, there are some methods that could benefit more from the proposed strategy, such as the reformulation of the Kernel Discriminant Analysis to ordinal regression (KDLOR) [2]. This method, which is the one chosen for the experimental part of the study, makes a very strong assumption when fixing the bias terms and does not include them in the optimisation of the model (while they are indeed a extremely important part of the model, that could lead to poor results when not optimised correctly). We propose to include both the parameters associated to the log-gamma distribution and the thresholds of the model in the proposed probability estimation optimisation step. In the next subsection, we will include some introductory notions of this method for the sake of understanding.

2.1 Discriminant Learning

We briefly introduce some notions of discriminant learning in this subsection. Its main objective is to find the optimal projection for the data (which allows the classes of the problem to be easily separated). To do so, the algorithm analyses two objectives: the maximisation of the between-class distances, and the minimisation of the within-class distances, by using variance-covariance matrices ($\mathbf{S}_b$ and $\mathbf{S}_w$ respectively) and the so-called Rayleigh coefficient ($J(\boldsymbol{\beta}) = \frac{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_b \boldsymbol{\beta}}{\boldsymbol{\beta}^{\mathrm{T}} \mathbf{S}_w \boldsymbol{\beta}}$, where $\boldsymbol{\beta}$ is the projection to be found). To achieve these objectives, the $K-1$ eigenvectors associated with the highest eigenvalues of $\mathbf{S}_w^{-1} \cdot \mathbf{S}_b$ are computed, and these will be the mapping functions which project the data to a lower-dimensional space, which will be used as the discriminant function. As stated before, this learning methodology has also been adapted to ordinal classification [2] by imposing a constraint on the projection to be computed, so that it will preserve and take advantage of the ordinal information. This constraint forces the projected classes to be ordered according to their rank, which is useful for minimising ordinal misclassification errors. This method is known as Kernel Discriminant Learning for Ordinal Regression (KDLOR). Further information can be found in [2].

2.2 Bias Computation for the Discriminant Function

The bias terms (both in the original binary Discriminant Analysis and the ordinal version) can be derived from the Bayes theorem [7], assuming that the projected patterns follow a normal distribution with equal variance and a priori

probabilities. That is,

$$P(y_i = \mathcal{C}_k | X = \mathbf{x}_i) = \frac{f_k(\mathbf{x}_i)\pi_k}{\sum_{l=1}^{K} f_l(\mathbf{x}_i)\pi_l} = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2\right)}, \quad (1)$$

where π_k is the prior probability of class k and $f_k(\mathbf{x}_i)$ the class-conditional density function of X for class y. Assuming that each class density function (f_k) can be modelled by a univariate Gaussian distribution (as done in the right part of Eq. 1), it can be seen that this is equivalent to assigning $\mathbf{x}_i$ to the class with the largest discriminant score (taking logs and discarding terms that do not depend on k):

$$\gamma_k(\mathbf{x}_i) = \mathbf{x}_i \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k), \quad (2)$$

where μ_k is the mean of $\boldsymbol{\beta}^{\mathrm{T}} X_k$ and σ the variance (assuming that all the variances for the classes are equal). For $K = 2$, the bias can be fixed to the point $\mathbf{x}$ where the discriminant scores for both classes are equal (decision boundary). This leads us to:

$$b = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2 \cdot \log(\pi_2)}{\mu_1 - \mu_2} - \frac{\sigma^2 \cdot \log(\pi_1)}{\mu_1 - \mu_2}. \quad (3)$$

In the case of KDLOR, a priori probabilities π_1 and π_2 are assumed to be equal (therefore, $b = \frac{\mu_1+\mu_2}{2}$). Moreover, each bias is computed considering the adjacent projected classes (e.g. b_1 is computed considering $\mathcal{C}_1$ and $\mathcal{C}_2$):

$$b_i = \frac{\mu_i + \mu_{i+1}}{2}. \quad (4)$$

In contrast, we propose to consider the full expression of (3), given that a priori probabilities are generally different, specially in the case of ordinal regression, where extreme classes are usually associated to infrequent events.

2.3 Maximum Likelihood Based Methodology

Instead of considering the technique introduced in previous subsection, we can perform a maximum likelihood estimation of the biases of the probability distributions, which will be introduced in this subsection. In order to do so, we will consider ordinal logistic regression models. The well-known binary logistic regression model can be easily generalised to handle an ordinal response [4,3], leading to cumulative link models (CLMs). Let h denote a given link function, then, the model:

$$h[(P(y_i \prec \mathcal{C}_j))] = b_j - \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_i, \;\; j = 1, \ldots, K-1, \;\; b_1 < \ldots < b_{K-1}, \quad (5)$$

links the cumulative probabilities to a linear predictor based on the parameter vector $\boldsymbol{\beta}$. By definition, $P(y_i \prec \mathcal{C}_K) = 1$. Let $F = h^{-1}$ denote the inverse link

function for the CLM (e.g. the normal cdf for the cumulative probit model). The log-likelihood function can be defined:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log[F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_j, q_j) - F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_{j-1}, q_{j-1})], \tag{6}$$

where, for $\mathbf{x}_i$, $y_{ij} = 1$ if $y_i = \mathcal{C}_j$ and $y_{ij} = 0$ otherwise, $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \mathbf{b}, \mathbf{q}\}$ is the vector of parameters of the model, $\mathbf{b}$ is the vector containing the biases and $\mathbf{q}$ is the vector of possible distribution parameters. Note that, by definition, $F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_0, q_0) = 0$ and $F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_K, q_K) = 1$. Therefore, the probability of a pattern $\mathbf{x}_i$ belonging to a given class $\mathcal{C}_j$ is computed as follows:

$$P(y_i = \mathcal{C}_j|\mathbf{x}_i) = F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_j, q_j) - F(\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i, b_{j-1}, q_{j-1}), \tag{7}$$

i.e. the difference of the cumulative probabilities.

There are multiple options for F [3]. However, in this paper we will consider the log-gamma function [6], which depends on a parameter q, generalising the log-log link function ($q > 0$), the probit link ($q = 0$) and the complementary log-log ($q < 0$). The log-gamma link can be written as follows:

$$F(z_i, b_j, q_j) = \begin{cases} 1 - \Gamma_{\mathrm{inc}}(q_j^{-2}, v_{ij}), & q_j < 0, \\ \Phi(b_j - z_i), & q_j = 0, \\ \Gamma_{\mathrm{inc}}(q_j^{-2}, v_{ij}), & q_j > 0, \end{cases} \tag{8}$$

where $v_{ij} \equiv q_j^{-2}\exp(q_j \cdot (b_j - z_i))$, $z_i = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_i$, $\Gamma_{\mathrm{inc}}(\cdot,\cdot)$ denotes the standardised incomplete gamma function $\Gamma_{\mathrm{inc}}(a,x) = \int_0^x \exp(-t)\cdot t^{a-1}dt \cdot \frac{1}{\Gamma(a)}$ and Φ corresponds to the standard normal distribution. Using the log-gamma link, the density is negatively skewed for $q < 0$, positively skewed for $q > 0$ and the absolute skewness and kurtosis increase monotonically in $|q|$ [6]. The cumulative and standard probabilities obtained using this function for different q values for a single class can be seen in Fig.1. On the contrary, Fig. 2 shows these probabilities for a problem with 4 ordered classes and different q values.

We will consider a nonlinear projection given by other ordinal regression model (in our case, KDLOR). Let z_i the projection of pattern $\mathbf{x}_i$, $g(\mathbf{x}_i) = z_i$. Consequently, the parameter vector will be $\boldsymbol{\theta} = \{\mathbf{b}, \mathbf{q}\}$. Because of the differentiability of the log-likelihood $\mathcal{L}(\{\mathbf{b}, \mathbf{q}\})$ with respect to the parameters $\mathbf{b}$ and $\mathbf{q}$, a gradient-ascent algorithm can be used to maximise it:

$$\{\mathbf{b}^*, \mathbf{q}^*\} = \arg\max_{\mathbf{b},\mathbf{q}} \mathcal{L}(\{\mathbf{b}, \mathbf{q}\}). \tag{9}$$

The gradient vector will be composed of partial derivatives $\nabla\mathcal{L} = \left[\frac{\partial\mathcal{L}}{\partial b_1}, \ldots, \frac{\partial\mathcal{L}}{\partial b_{K-1}}, \frac{\partial\mathcal{L}}{\partial q_1}, \ldots, \frac{\partial\mathcal{L}}{\partial q_{K-1}}\right]$. Note that the optimised parameters by our proposal are the bias terms $\mathbf{b}$ and the parameters $\mathbf{q}$ associated to the link function F (but not the projection model $\boldsymbol{\beta}$).

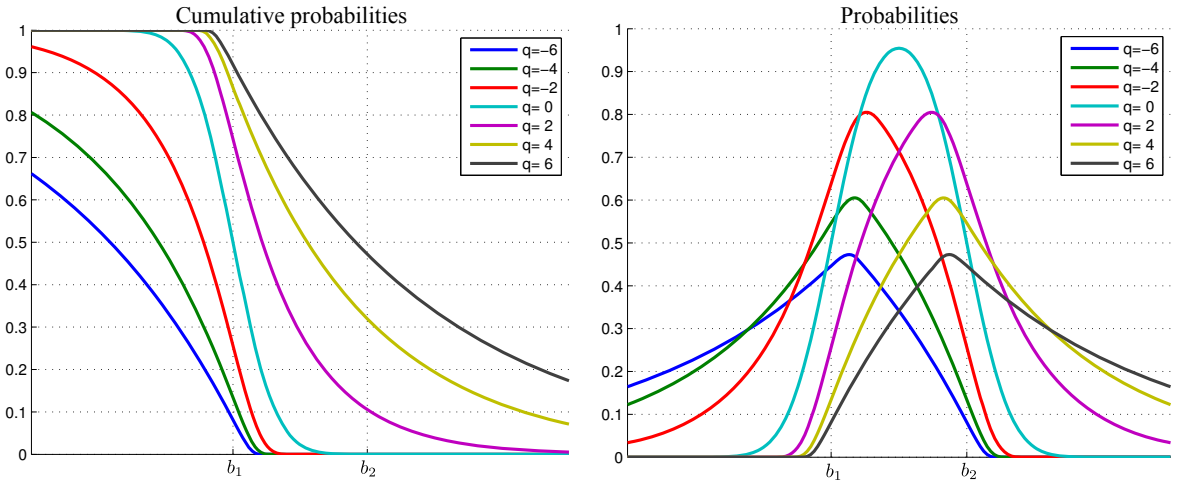


Fig. 1. Log-gamma probabilities for given b_1 and b_2 and different q values

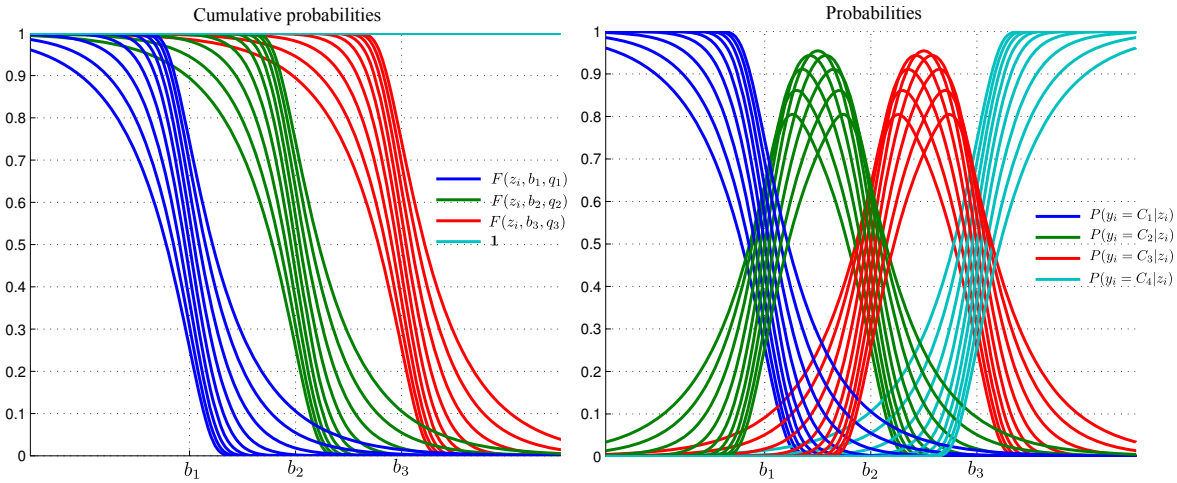


Fig. 2. Log-gamma computed probabilities for a 4-class problem and different q values

The derivatives of the likelihood with respect to the vector **b** and **q** are:

$$\frac{\partial \mathcal{L}}{\partial b_j} = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \frac{\delta_{jk} \cdot \frac{\partial F}{\partial b_j}(z_i, b_j, q_j) - \delta_{j-1,k} \cdot \frac{\partial F}{\partial b_j}(z_i, b_{j-1}, q_{j-1})}{F(z_i, b_j, q_j) - F(z_i, b_{j-1}, q_{j-1})}, \tag{10a}$$

$$\frac{\partial \mathcal{L}}{\partial q_j} = \sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij} \frac{\delta_{jk} \cdot \frac{\partial F}{\partial q_j}(z_i, b_j, q_j) - \delta_{j-1,k} \cdot \frac{\partial F}{\partial q_j}(z_i, b_{j-1}, q_{j-1})}{F(z_i, b_j, q_j) - F(z_i, b_{j-1}, q_{j-1})}, \tag{10b}$$

where f denotes the derivative of F, i.e. the probability density function corresponding to the cumulative density function F, and δ_{jk} is the Kronecker delta, i.e. $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ otherwise.

If $q_j \neq 0$, the derivative of F with respect to b_j:

$$\frac{\partial F}{\partial b_j} = \frac{q_j \cdot \exp(-z) \cdot r_{ij}^{q_j^{-2}}}{\Gamma(q_j^{-2})}, \tag{11}$$

where, for the sake of simplicity, we denote $r_{ij} = \frac{\exp(q_j \cdot (b_j - z_i))}{q_j^2}$.

If $q_j \neq 0$, the derivative of F with respect to q_j is:

$$\frac{\partial F}{\partial q_j} = \frac{1}{q_j^3 \cdot \Gamma(q_j^{-2})} \exp(-r_{ij}) \Bigg(2\exp(r_{ij}) \left(G^{\,3\,0}_{2\,3} \left(\begin{matrix} 0,0 \\ 0,0,q_j^{-2} \end{matrix} \middle| r_{ij} \right) \right) + \\ + q_j^2 (b_j q_j - q_j z_i - 2) r_{ij}^{q_j^{-2}} + 2\exp(r_{ij}) \Gamma(q_j^{-2}, r_{ij}) \Big(\log(r_{ij}) - \psi^{(0)}(q_j^{-2}) \Big) \Bigg), \quad (12)$$

where $q_j \neq 0$, $G^{\,m\,n}_{p\,q} \left(\begin{matrix} a_1,\ldots,a_p \\ b_1,\ldots,b_q \end{matrix} \middle| z \right)$ it the Meijer G-function [8] and $\psi^{(n)}(\cdot)$ is the n-th derivative of the digamma function.

Finally, for the case $q = 0$, the derivatives are:

$$\frac{\partial F}{\partial b_j} = \frac{\exp\left(-\frac{(b_j - z_i)^2)}{2} \right)}{\sqrt{2\pi}}, \quad \frac{\partial F}{\partial q_j} = 0. \quad (13)$$

In this work, the iRprop$^+$ algorithm is used to optimise the likelihood, because of its proven robustness [9]. Therefore, each parameter b_i and q_i will be updated considering the sign of the derivative but not the magnitude. Although the second partial derivatives can also be computed and used for optimisation, they could actually make this process more computationally costly due to the complexity of the associated formula.

A possible result for the proposed optimisation methodology can be seen in Fig. 3 for a 4-class ordinal problem. It can be seen that considering different q values for the different classes results in a more complex model, where the maximum probability class between a pair of thresholds is not always the one corresponding to the first threshold (contrary to standard CLMs such as the POM model).

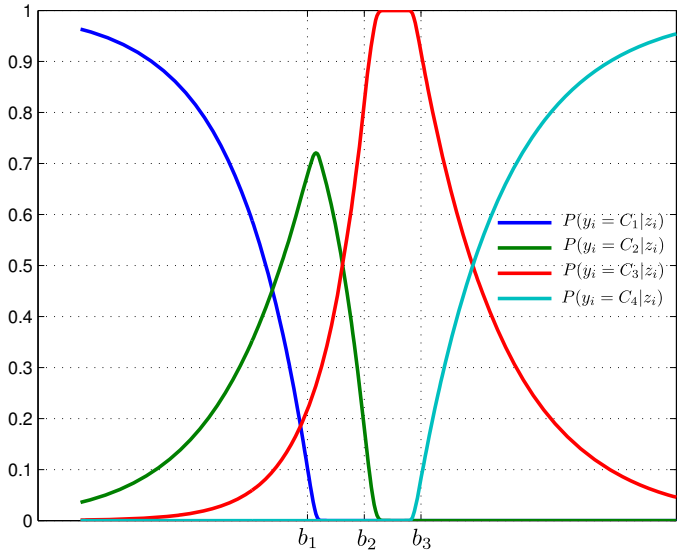


Fig. 3. Probabilistic distributions obtained for eucalyptus and ESL

Table 1. Characteristics of the benchmark datasets in alphabetical order

Dataset	#Pat.	#Attr.	#Classes	Class distribution
balance-scale	625	4	3	(288, 49, 288)
contact-lenses	24	6	3	(15, 5, 4)
ESL	488	4	9	(2, 12, 38, 100, 116, 135, 62, 19, 4)
eucalyptus	736	91	5	(180, 107, 130, 214, 105)
LEV	1000	4	5	(93, 280, 403, 197, 27)
pasture	36	25	3	(12, 12, 12)
squash-stored	52	51	3	(23, 21, 8)
SWD	1000	10	4	(32, 352, 399, 217)
tae	151	54	3	(49, 50, 52)
toy	300	2	5	(35, 87, 79, 68, 31)

The original prediction decision rule for the KDLOR method was the following: $y_i^* = \mathcal{C}_j$ if $b_{j-1} < z_i < b_j$, where $b_0 \equiv -\infty$ and $b_K \equiv \infty$. However, in this case it should be done considering the class associated to the maximum probability computed using the log-gamma function:

$$y_i^* = (C_j | j = \arg\max_j (F(b_j - z_i, q_j) - F(b_{j-1} - z_i, q_{j-1}))). \tag{14}$$

3 Experiments

Several benchmark datasets with different characteristics have been tested in order to validate the methodology proposed. Table 1 shows the characteristics of these datasets, where the number of patterns, attributes, classes and the class distribution (number of patterns per class) can be seen.

The experiments were designed to compare three different methodologies. First of all, the KDLOR nonlinear projection is obtained and then the biases (and the parameters of the distributions) are learnt using one of the following methods:

- The original KDLOR method considering the methodology in Eq. (4) for setting the thresholds and assuming equal a priori probabilities (KDLOR).
- KDLOR considering the complete expression of Eq. (3) for setting the thresholds without assuming equal a priori probabilities (AP-KDLOR).
- KDLOR optimising the bias and the distribution parameters via maximum likelihood (ML-KDLOR) (see Section 2.3), with the parametrised log-gamma function.

3.1 Evaluation Metrics

Several measures can be considered for evaluating ordinal classifiers, e.g. the mean absolute error (MAE) and the well-known accuracy (Acc) [1,2]. While the

Acc measure is also intended to evaluate nominal classifiers, the MAE metric is the most common choice for ordinal methods.

The mean absolute error (MAE) is the average deviation in absolute value of the predicted class from the true class [10]: $MAE = \frac{1}{N}\sum_{i=1}^{N} e(\mathbf{x}_i)$, where $e(x_i) = |r(y_i) - r(y_i^*)|$ is the distance (in number of categories) between the true and the predicted ranks. $r(y)$ is the rank for a given target y (its position in the ordinal scale), so MAE values range from 0 to $K-1$ (maximum deviation in number of ranks between two labels).

3.2 Experimental Setting

Regarding the experimental setup, a holdout stratified technique was applied to divide the datasets 30 times, using 75% of the patterns for training and the remaining 25% for testing. The parameters of each algorithm are chosen using a nested validation for each of the training sets (k-fold method with $k = 3$) and the validation criteria is the MAE error (see Section 3.1), since it can be considered the most common one in ordinal regression. The kernel selected for KDLOR is the Gaussian one, $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)$ where σ is the standard deviation and is cross-validated within the following values: $\{10^{-3}, \ldots, 10^3\}$.

As stated before, the optimisation of gradient-based methods is guaranteed only to find a local minimum; therefore, the quality of the solution can be sensitive to initialisation. The initial $\mathbf{b}$ value for the gradient-ascent was set to the original biases of the KDLOR method (Eq. (4)). The $\mathbf{q}$ vector associated to the log-gamma distribution parameters were randomly initialised between $[-1, 1]$ (recall that the thresholds were firstly computed assuming a normal distribution). The gradient norm stopping criterion was set at 10^{-8} and the maximum number of conjugate gradient steps at 10^2 [9].

3.3 Results

Table 2 shows the mean test results for the 10 datasets considered in terms of *Acc* and MAE. First of all, it can be appreciated from this Table that the results for KDLOR and AP-KDLOR are very similar. This indicates that the consideration of the a priori probabilities for the bias computation does not influence the results to a great extent and therefore it can be assumed that these are equal. On the other hand, one can appreciate that the optimisation via maximum likelihood results in a better performance of the algorithm in both metrics (specially if ones takes into account that the projections z_i remain unchanged and that the improvement is only due to the optimisation of the thresholds and distribution parameters). However, there are also some cases in which the capability of the proposal is limited (e.g. for the tae dataset, where the same results are obtained for the three methods).

Table 2. Mean test results for *Acc* and *MAE*. Best results are highlighted in boldface

Dataset	Method	*Acc*	*MAE*
balance-scale	KDLOR	82.55 ± 3.54	0.176 ± 0.040
	AP-KDLOR	82.55 ± 3.54	0.176 ± 0.040
	ML-KDLOR	**90.70 ± 0.72**	**0.103 ± 0.014**
contact-lenses	KDLOR	61.11 ± 18.74	0.533 ± 0.257
	AP-KDLOR	61.11 ± 18.74	0.533 ± 0.257
	ML-KDLOR	**65.00 ± 12.65**	**0.506 ± 0.212**
ESL	KDLOR	64.34 ± 3.41	0.374 ± 0.040
	AP-KDLOR	64.34 ± 3.41	0.374 ± 0.040
	ML-KDLOR	**68.47 ± 2.85**	**0.330 ± 0.033**
eucalyptus	KDLOR	59.91 ± 2.70	0.450 ± 0.037
	AP-KDLOR	59.91 ± 2.70	0.450 ± 0.037
	ML-KDLOR	**61.01 ± 2.78**	**0.437 ± 0.035**
LEV	KDLOR	55.12 ± 2.81	0.507 ± 0.033
	AP-KDLOR	55.12 ± 2.81	0.507 ± 0.033
	ML-KDLOR	**60.49 ± 2.73**	**0.436 ± 0.031**
pasture	KDLOR	61.85 ± 11.56	0.385 ± 0.119
	AP-KDLOR	61.69 ± 11.72	0.387 ± 0.121
	ML-KDLOR	**62.22 ± 11.15**	**0.381 ± 0.116**
squash-stored	KDLOR	62.86 ± 14.10	0.387 ± 0.150
	AP-KDLOR	**63.08 ± 13.76**	0.385 ± 0.147
	ML-KDLOR	**63.08 ± 15.17**	**0.377 ± 0.161**
SWD	KDLOR	48.87 ± 3.00	0.579 ± 0.036
	AP-KDLOR	48.87 ± 3.00	0.579 ± 0.036
	ML-KDLOR	**56.65 ± 4.08**	**0.462 ± 0.046**
tae	KDLOR	**56.67 ± 5.30**	**0.452 ± 0.055**
	AP-KDLOR	**56.67 ± 5.30**	**0.452 ± 0.055**
	ML-KDLOR	**56.67 ± 5.30**	**0.452 ± 0.055**
toy	KDLOR	88.67 ± 3.35	0.113 ± 0.034
	AP-KDLOR	88.67 ± 3.35	0.113 ± 0.034
	ML-KDLOR	**90.91 ± 2.56**	**0.092 ± 0.026**

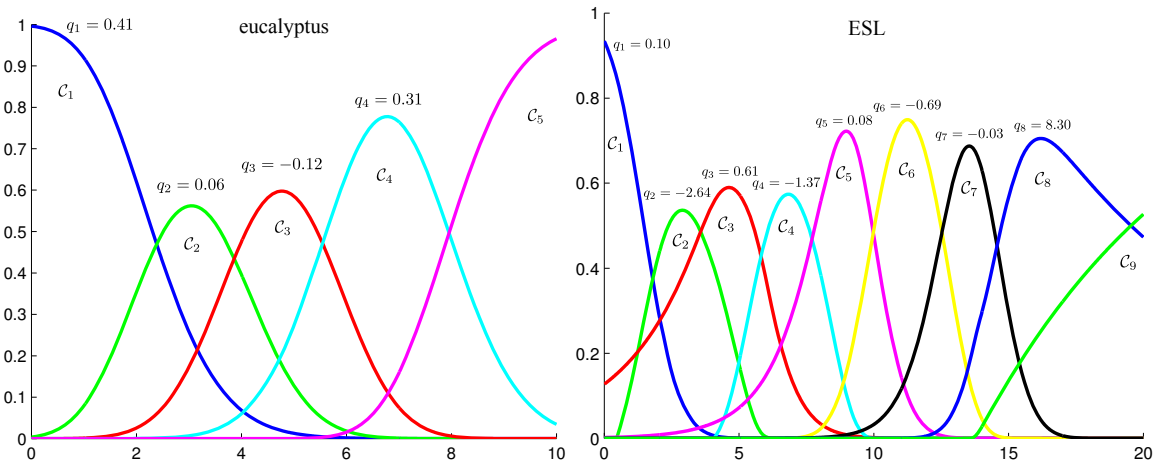


Fig. 4. Probabilistic distributions obtained for eucalyptus and ESL

Furthermore, it is important to note that the proposed methodology consider the optimisation of the thresholds and distribution parameters as a whole, taking into account all the classes in the problem. This is opposed to the bias computation used for KDLOR and AP-KDLOR, where these are computed considering adjacent classes only. To understand how this could influence the results in terms of the classes ordering, we can check the results for the squash-stored dataset, where both AP-KDLOR and ML-KDLOR obtained the same performance in *Acc*, while AP-KDLOR presented worse results for MAE.

Fig. 4 shows the resultant probability distributions for two of the considered datasets. It can be seen that the assumption of $q = 0$ for all probability distributions may not hold, and performing the proposed maximum likelihood optimisation results in a much higher flexibility (which also improves the generalisation performance, see Table 2).

4 Conclusions

In the context of ordinal regression threshold models, this paper proposes a gradient-ascent optimisation of the log-gamma distribution via maximum likelihood to obtain more accurate probabilistic predictions. The experimental results show a good synergy between the proposed technique and the reformulation of the well-known kernel discriminant analysis to ordinal regression problems. More specifically, the results suggests that a per-class bias and probability distribution optimisation is indeed a crucial step for the base methodology, leading to an improvement of the performance. As future work, the methodology proposed in this paper could be used in conjunction with some probabilistic ensemble methods for ordinal regression [11] or derived for and included in other statistical and ordinal link-based methodologies [4].

References

1. Gutiérrez, P.A., Pérez-Ortiz, M., Fernández-Navarro, F., Sánchez-Monedero, J., Hervás-Martínez, C.: An Experimental Study of Different Ordinal Regression Methods and Measures. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part II. LNCS, vol. 7209, pp. 296–307. Springer, Heidelberg (2012)
2. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel discriminant learning for ordinal regression. IEEE Transactions on Knowledge and Data Engineering 22, 906–910 (2010)
3. Agresti, A.: Analysis of ordinal categorical data. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley (1984)
4. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society 42(2), 109–142 (1980)
5. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural Computation 19, 792–815 (2007)
6. Lin, K.C.: Goodness-of-fit tests for modeling longitudinal ordinal data. Comput. Stat. Data Anal. 54(7), 1872–1880 (2010)

7. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction, 2nd edn. Springer (2008)
8. Askey, R.A., Daalhuis, A.B.O.: Generalized Hypergeometric Functions and Meijer G-Function. In: NIST Handbook of Mathematical Functions, pp. 403–418. Cambridge University Press (2010)
9. Igel, C., Hüsken, M.: Empirical evaluation of the improved rprop learning algorithms. Neurocomputing 50, 105–123 (2003)
10. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 2009), Pisa, Italy (2009)
11. Pérez-Ortiz, M., Gutiérrez, P.A., Hervás-Martínez, C.: Projection based ensemble learning for ordinal regression. IEEE Transactions on Cybernetics (99) (2013), http://dx.doi.org/10.1109/TCYB.2013.2266336 (accepted)

I am incapable of conceiving infinity, and yet I do not accept finity.

Simone de Beauvoir

4

Kernel functions and ordinal kernel learning

This chapter comprises some contributions of the thesis related to the topic of kernel learning, including an analysis of the methods that could be used for optimising multi-scale kernels and a new ordinal kernel learning algorithm. Moreover, a new method for kernelising any ordinal learning algorithm is also derived, as well as a strategy for introducing privileged information in the kernel matrix.

Main publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. A study on multi-scale kernel optimisation via centred kernel-target alignment. *Neural Processing Letters* (Under Review), 2014, Impact Factor (2013): 1.237 (Q2).
- M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelising the Proportional Odds Model through Kernel Learning techniques. *Neurocomputing*, In press, 2014, Impact Factor (2013): 2.005 (Q1).
- M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Incorporating privileged information to improve manifold ordinal regression. In *International Conference on Neural Computation Theory and Applications*, pages 187–194, 2014.

Other publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez, J. Sánchez-Monedero and C. Hervás-Martínez. Multi-scale support vector machine optimization by kernel-target alignment. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), pages 391–396, 2013.
- M. Pérez-Ortiz, P.A. Gutiérrez, M. Cruz-Ramírez, J. Sánchez-Monedero and C. Hervás-Martínez. Kernelizing the proportional odds model through the empirical kernel mapping. In *International Work Conference on Artificial Neural Networks* (IWANN), Lectures Notes in Computer Science Volume 7902, pages 270–280, 2013.
- M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Learning kernel label decompositions for ordinal classification problems. In *International Conference on Neural Computation Theory and Applications*, pages 218–225, 2014.

The three main publications are now presented in the different subsections of this chapter.

4.1. A study on multi-scale kernel optimisation via centred kernel-target alignment

The crucial ingredient of kernel methodologies is undoubtedly the application of the so-called kernel trick, a procedure which maps the data into a higher-dimensional, or even infinite, feature space. The data separation in this space is proved to be easier, allowing the formulation of nonlinear variants of any algorithm which can be cast in terms of the inner products between data points. In this line, with the aim of better fitting the data, different kernels and different optimisation strategies have been proposed.

The following paper considers the problem of optimising a multi-scale kernel and reviews the approaches that have been considered in the literature. This type of kernels have received very few attention from the machine learning community, although they have been proven to perform well in the presence of heterogeneous attributes. This could be due to the large number of parameters in this and other kernel formulations, which precludes the application of a traditional cross-validation procedure.

The paper selects one of the most outstanding approaches to kernel learning (centred kernel-target alignment or CKTA) and reformulates this technique to work with multi-scale kernels via a gradient-descent scheme. We also compare this and other alternatives and provide some clues and insights into the usefulness of CKTA. The results of the experiments in this paper show that the use of CKTA for optimising a multi-scale kernel leads to the construction of a well-defined feature space and simpler classification models. Furthermore, this method is also seen to filter non-informative features and achieve robust

results. Finally, some considerations about when a multi-scale kernel could be useful are given and a distance-based initialisation technique for the gradient-descent is presented. The good results obtained in this paper for nominal classification problems encourage us to apply KTA techniques for ordinal regression.

Neural Processing Letters manuscript No.
(will be inserted by the editor)

A study on multi-scale kernel optimisation via centred kernel-target alignment*

M. Pérez-Ortiz · P.A. Gutiérrez · J. Sánchez-Monedero · C. Hervás-Martínez

Received: date / Accepted: date

Abstract Kernel mapping is one of the most widespread approaches to intrinsically deriving nonlinear classifiers. With the aim of better suiting a given dataset, different types of kernels have been proposed and different bounds and methodologies have been studied to optimise these kernels. We focus on the optimisation of a multi-scale kernel, where a different width is chosen for each feature. This idea has been barely studied in the literature, although it has been shown to achieve better performance in the presence of heterogeneous attributes. The large number of parameters in multi-scale kernels makes it computationally unaffordable to optimise them by applying traditional cross-validation. Instead, an analytical measure known as centred kernel-target alignment (CKTA) can be used to align the kernel to the so-called ideal kernel matrix. This paper analyses and compares this and other alternatives, providing a review of the state-of-the-art in kernel optimisation and some insights into the usefulness of multi- scale kernel optimisation via CKTA. When applied to the binary support vector machine paradigm (SVM), the results using 24 datasets show that CKTA with a multi-scale kernel leads to the construction of a well-defined feature space and simpler SVM models, provides an implicit filtering of non-informative features and achieves robust and comparable performance to other state-of-the-art methods even when using random initialisations. Finally, we derive some considerations about when a multi-scale approach could be, in general, useful and propose a distance-based

* This paper is a very significant extension of a preliminar conference version [43] including much additional material: a comprehensive review of kernel model selection methods, a more detailed description of the method considered, and a wider experimental section, comparing to other multi-scale algorithms, and using a wider set of benchmark datasets. Besides, some hints about when multi-scale kernels are useful and how to initialise them are provided.

Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071, Córdoba, Spain. Tel.: +34 957 218 349; Fax: +34 957 218 630.
E-mail addresses: {i82perom, pagutierrez, jsanchezm, chervas}@uco.es

initialisation technique for the gradient-ascent method, which shows promising results.

Keywords kernel-target alignment · kernel methods · multi-scale kernel · parameter selection · support vector machines · cross-validation

1 Introduction

The crucial ingredient of kernel methodologies is undoubtedly the application of the so-called *kernel trick* [56], a procedure that maps the data into a higher-dimensional, or even infinite, feature space $\mathcal{H}$ via some mapping Φ. This allows the formulation of nonlinear variants of any algorithm that can be cast in terms of inner products between data points. Instead of explicitly computing the function Φ, $\mathcal{H}$ can be efficiently obtained from a suitable kernel function. Indeed, this kernel function implicitly determines the feature space $\mathcal{H}$ in such a way that a poor choice of this function can lead to significantly impaired performance. These choices are related to the definition of a metric between input patterns that fosters correct classification. Usually, a parametrised set of kernels is considered for this purpose, although it is still necessary to choose a performance measure and an optimisation strategy. This optimisation is often performed using a grid-search or cross-validation procedure over a previously defined search space.

Some authors suggest the use of the multi-scale kernel [10] (also known as a multi-parametric, anisotropic or ellipsoidal kernel), where a different kernel parameter is chosen for each feature. The general motivation for the use of multi-scale kernels is that in real-world applications the attributes can present very different natures, which hampers the performance of spherical kernels (i.e., with the same kernel width for each attribute). It is clear that more flexible kernels could fit heterogeneous datasets better, leading to a lower generalisation error [32, 22]. However, the number of parameters (i.e., as many as the number of features) makes the computational cost prohibitive when considering a cross-validation technique. For this reason, these kernels have been barely used in the literature; when they have been used, they have been optimised by evolutionary algorithms [24, **?**, 22] or by gradient-based techniques applied to some measure of kernel quality or generalisation error bound [48, 10, 32]. To find a suitable hyperparameter optimisation technique for multi-scale kernels, this paper first reviews the alternative methodologies that have been previously proposed for single-parameter kernel optimisation.

Ideally, we would like to find the kernel that minimises the true risk of a specific classifier for a specific dataset. Unfortunately, this quantity is not accessible; therefore, different estimates or bounds have been developed based on both analytical and experimental knowledge, such as the span of support vectors [55] or the radius margin bound [56]. This problem has also been tackled using evolutionary algorithms [17, 31, 44], meta-learning approaches [50], or Bayesian inference [52], by defining data-dependent estimates of the complexity of a function class [38, 4] or simply by optimising the class separability

in the feature space [60]. In most of these cases, a large amount of computation time is needed because the bounds or the algorithms require training the learning machine several times and might even require solving an additional optimisation problem. Moreover, some of the bounds are not differentiable, which means that they must be smoothed to use a gradient descent method [10], which can result in a loose solution for the problem that is tackled.

To overcome these handicaps, a differentiable and simpler approach has been proposed, which is known as kernel-target alignment (KTA) [14,11]. KTA is independent of the learning algorithm, and it thus avoids the expensive computational procedure of training the classifiers. Essentially, KTA aims to find a kernel function k in a restricted family of kernels such that the induced Gram matrix presents the smallest distance to the ideal kernel matrix, which preserves perfectly the entire training label structure (represented in this case by similarities between patterns). Moreover, several optimisation strategies have been developed in the literature to maximise KTA, such as greedy algorithms [15], quadratically constrained quadratic programs, linearly constrained quadratic programs (QP) [11] or multiple kernel learning problems [39]. Centred KTA (CKTA) [11] is an extension of KTA that has recently been shown to correlate better with peformance and to avoid some data distribution problems related to KTA.

The first objective of this paper is to provide an analysis of the state-of-the-art in kernel optimisation to find the most appropriate method for the multi-scale kernel. As a result of this analysis, several advantages of CKTA have been identified over the rest of the methods: algorithm independence, data distribution independence and simple optimisation. Therefore, this paper considers CKTA to select the multiple parameters of multi-scale kernels (multi-scale centred kernel-target alignment, MSCKTA). The measure is optimised by a gradient ascent procedure in which the free parameters are the different kernel widths of each feature, which, as we will show, leads inherently to the filtering of non-informative features. To the best of the authors' knowledge, this idea has been considered only in [32] and [26]. In the former one, non-centred KTA is used to optimise a multi-scale version of a special type of kernel for the analysis of biological sequence data, i.e., oligo kernels. In the latter, non-centred KTA is also tested to compare spherical and multi-scale kernels with different optimisation techniques. In the case of [26], although it is not clear that a multi-scale kernel may be in general useful, the author argues that KTA is clearly the best suited method for model selection in high-dimensional search spaces. The experiments performed in this paper include a more general experimental setup with 24 benchmark datasets and statistical comparisons to other uni and multi-scale state-of-the-art methodologies, comprising an extensive experimental analysis that has not been performed until now in the context of multi-scale kernels. Moreover, we also propose a novel deterministic distance-based strategy for initialising the coefficient vector for the gradient-ascent algorithm, which is compared to random and fixed initialisations. The results suggest that MSCKTA is a robust technique that provides binary SVM with a higher flexibility to address heterogeneous real datasets

and a better determined feature space that results in simpler SVM models (in terms of the number of support vectors) at a reasonable computational complexity. This additional complexity when compared to uni-scale methods is the price to pay to obtain more accurate and simpler models. These conclusions are reinforced by graphically analysing those datasets in which the performance is significantly improved by MSCKTA, thus providing some hints about when the method should be applied. Furthermore, as said, the methodology naturally spans a feature filter which could be beneficial for model interpretation purposes.

The rest of the paper is organized as follows: Section II shows the state-of-the-art in kernel optimization for completeness and analyses what methods are better suited for multi-scale kernels; Section III presents the MSCKTA optimization method; Section IV describes the experimental study and analyses the results obtained; and Section V outlines some conclusions and future work.

2 Related research

This section establishes the terminology and notation that will be used throughout this study and reviews the methodologies in the state-of-the-art literature. The goal in binary classification is to assign an input vector $\mathbf{x}$ to one of $\{+1, -1\}$ classes (this label will be designed as y, where $y \in \mathcal{Y} = \{-1, +1\}$), when considering an input space $\mathcal{X} \in \mathbb{R}^d$, where d is the data dimensionality. The training data are assumed to be generated from an i.i.d. $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ from an unknown distribution $P(\mathbf{x}, y)$. Therefore, the objective in this type of problem is to find a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y},\ f \in \mathcal{F}$ that minimises the expected loss or risk [56].

The methods presented in this paper will be applied to the binary SVM paradigm [5,12] because of the proven performance of the binary SVM classifier and the large amount of work that has been accomplished on kernel selection for this methodology.

As is well-known, the SVM algorithm depends on several parameters. On the one hand, the cost parameter C controls the trade-off between margin maximisation and error minimisation. On the other hand, kernel parameters appear in the non-linear mapping into the feature space. In the following subsections, we analyse how these parameters (in the context of kernel optimisation) have been properly adjusted in the literature, providing a taxonomy that divides the analysis into the following parts:

1. Definition of the problem to solve, distinguishing between the kernel function selection, kernel parameter optimisation and multiple kernel learning.
2. Presentation of different estimators that have been previously proposed in the literature, dividing them in two categories: algorithm-dependent methods (which require explicit training of the kernel machine) and algorithm-independent (which do not consider any concrete learning algorithm).

3. Presentation of the optimisation strategies used to optimise these estimators.
4. Analysis of the application of all of the techniques to the multi-scale kernel case.

2.1 Type of problem

In the same vein as the *No free lunch* theorem [59], which states that in the absence of prior information about the data no learning algorithm should be preferred, lies the notion of *No free kernel* [14], which considers the inexistence of a domain-independent kernel, i.e., a universal kernel that performs perfectly in all types of situations. In fact, choosing the most appropriate kernel depends on the problem at hand. Thus, some studies have proposed methods for automatically selecting the best kernel function from a list of predefined functions [1,31]. Moreover, multiple kernel learning [27] is a different paradigm that is aimed at finding the best combination of a predefined list of kernels instead of using a single kernel. These different kernels can correspond to the use of different notions of similarity or information from multiple sources (e.g., different kernel functions, different hyperparameter values or even different feature subsets).

Both previously defined problems are outside the scope of this paper. The methods that we will present consider a specific kernel function (the Gaussian kernel), and they attempt to optimise the parameters associated with it such that the kernel fits the data in a better manner. This selection of hyperparameters is also a crucial step because it can drastically degrade or improve the performance. The number or nature of the parameters to optimise determines the type of optimisation problem to solve. For the case of multi-scale or ellipsoidal Gaussian kernels, the optimisation involves adjusting a vector of parameters. This paper will address precisely this type of problem.

2.2 Algorithm-dependent estimators for model selection

The methods analysed in this subsection are all dependent on the kernel machines considered, such that the solution for a kernel method would not be equally valid for a different kernel machine. The most widely used approach is the cross-validation method (CV), although it is a general procedure for parameter tuning and was not specifically designed for kernel methods. This methodology is based on the estimation of the expected error when the method is applied to an independent set of samples, which have not been used for training, and a grid-search is conducted through all of the parameter combinations. Usually, for this purpose, the training data are split into several subsets, and the prediction error is estimated as the mean of all of the prediction errors when using each subset for testing. Finally, the parameters are chosen as the parameters that perform better according to specific criteria, and the training

process must be repeated using the whole set of training data. Although CV is a reliable estimator, it presents an important computational load because it implies the execution of the algorithm on every possible value of the parameter vector (up to some discretisation). Furthermore, as a step forward, previous research [36] has presented a gradient-based methodology that uses a smooth estimation of the validation function with respect to the SVM parameters.

Leave-one out (LOO) validation is also widespread in the literature because it provides an almost unbiased estimate of the error on the test data. It corresponds to an extreme case of CV in which the number of sets is equal to the number of training points (i.e., the training is repeated as many times as training points, each time leaving one sample out of the training set to later test the algorithm). The computational cost in this case is even higher than for the CV. Given the high computational cost of the LOO validation, different strategies have been considered in the literature to provide an upper bound for the error or to approximate it using an expression that will be easier to compute. These strategies are focused on the specific case of SVMs and allow the optimisation of the kernel parameters. Some of them include the span of support vectors [55], the Jaakkola-Haussler bound [34], the Opper-Winther bound [41] or the Wahba's bound [58]. From these cases, we will focus our study in the radius margin bound and the span of support vectors.

These bounds are related to the concept of Empirical Risk Minimisation (ERM). Related to this concept, Vapnik and Chervonenkis introduced a measure of complexity of a class of functions $\mathcal{F}$, the VC (Vapnik-Chervonenkis) dimension [57], which is defined as the maximum number of points that can be learnt exactly by a function of $\mathcal{F}$. This concept was later characterised for the case in which $\mathcal{F}$ corresponds to a set of hyperplanes in $\mathbb{R}^d$ [56] (the VC dimension of $\mathcal{F}$ is $d+1$). From this consideration, a bound on the risk R of any function $f \in \mathcal{F}$ of VC dimension h and especially the one minimising the empirical risk R_{emp} was derived (the radius margin bound). This bound arose from the notion that the margin itself can not describe well how good a kernel is due to the negligence of the scaling. Indeed, it has been shown [56,3] that the capacity of such algorithms is bounded by $\frac{R^2}{M^2}$, where R is the radius of the smallest sphere enclosing the training points and M is the margin obtained on the training points. Radius margin bound was conceived to obtain an upper bound on the number of errors of the LOO procedure. The step function to optimise was not differentiable, which precluded the application of a gradient descent method. The number of scientific contributions that use this bound is very significant [13,18,23,10,19]. Nonetheless, ERM is considered to be an ill-posed problem (i.e., a slight change in the training set can entail a large change in the function); thus, several studies have focused on restricting the class of functions by imposing a regularisation constraint [25, 20], which implies that instead of minimising the empirical risk, one minimises the regularised risk. Furthermore, with regard to the VC dimension, certain data-dependent estimates of the complexity of a function class have been defined, as the Rademacher or Gaussian complexities [38,4] to derive general risk bounds for different types of algorithms.

The span of support vectors is based on the fact that Lagrange multipliers are adapted to accommodate the SVM threshold and to consider only a subset of the patterns (the patterns selected as support vectors). Based on this concept, [55] developed the span-rule to approximate the LOO error, which not only provides a good functional for SVM hyperparameter selection but also reflects the error better. However, this bound is very expensive to compute. Further work could give bounds with a high probability using the stability concept introduced in previous research [6].

Another branch of the parameter estimation techniques (which will later be used in comparisons) is based on the use of Bayesian methods [52,51] to tune the hyperparameters by maximising the so-called evidence (type-II likelihood) and obtaining predictive class probabilities rather than the conventional deterministic class label predictions.

2.3 Algorithm-independent estimators for model selection

As mentioned above, this subsection explores the kernel optimisation techniques that do not depend on the learning machine itself. This concept avoids the computational cost of training the algorithm and results in a solution that could be plugged into different learning machines. To accomplish these goals, different strategies are considered, such as the ideal kernel or the inter-cluster separability in the feature space induced by the kernel function.

The notion of ideal kernel has been extensively described and studied [14] where kernel-target alignment (KTA) was first proposed. This study was followed by a large amount of scientific contributions related to this estimator [11,15,30,45,32]. The *ugly duckling* theorem [14] informally states that an ugly duckling is just as similar to a swan as two swans are to each other without any sort of prior knowledge about the problem. As a consequence, any two arbitrary patterns are equally similar unless domain knowledge is used. In addition, the *luckiness framework* proposes better use of the information provided by the sample. Motivated by this purpose of extracting direct knowledge from data, the notion of KTA arises from the definition of an ideal kernel matrix that perfectly maintains the labelling structure [14]. Therefore, KTA focuses and emphasises supporting the information that is inherent to the data to perform the optimal mapping to the feature space (regardless of the algorithm to be employed)[1].

In [21], the notion of ideal kernel was studied by using three different measures of similarity among the matrices (KTA, the Frobenius distance and the correlation). These measures are applied to the optimisation of a spherical kernel on two different datasets. The results of comparing the traditional CV and these three methods show that the performance is similar, but KTA requires lower computational cost than the others.

The concept of distance metric learning has also been used for this purpose [39], by searching for a suitable linear map (i.e., performing a type of

[1] The KTA measure will be formally defined in Section 3.1

feature weighting) in the feature space, which computationally leads to a local-optima-free quadratic programming problem for the SVM case. In [60], the inter-cluster distances in the feature space are used to choose the kernel parameters, which therefore involves much less computation time than training the corresponding SVM classifiers and is competitive with the standard CV technique.

Other algorithm-independent kernel optimisation strategies have been developed in the literature. For example, the work in [61] is aimed at maximising the class separability in the empirical feature space, which is Euclidean and isomorphic to the original feature space.

2.4 Optimisation strategies

Once the kernel function (with a set of parameters to optimise) and the estimator (algorithm-dependent or algorithm-independent) has been selected, one must choose an optimisation technique.

As said, for the CV estimator, an exhaustive search is usually performed over the set of parameters, although several strategies have been proposed to guide this search (e.g., search space reductions [42], zoom search or genetic approaches [17,31,44]).

In general, when working with estimator functions, the most common approach is to use a gradient descent method, provided that the function is differentiable [10,36,45,21,48,46,7,18,23,13,62]. We will consider this method for the multi-scale approach used in this paper.

Concerning the aforementioned bounds that are specifically designed for SVMs, the optimisation is usually performed using a minimax method with a gradient descent approach, i.e., maximising the margin over the hyperplane and minimising a specific estimate of the generalisation error over the parameters [10]. Given that these methods are specific to SVMs, the cost parameter C is usually included in this optimisation process. A common trick is to consider C also as a kernel parameter, using the soft margin SVMs with quadratic penalisation of errors ($L2$-norm) and a modified kernel matrix ($\mathbf{K}$) [12]:

$$\mathbf{K} \leftarrow \mathbf{K} + \frac{1}{C}\mathbf{I}, \tag{1}$$

where $\mathbf{I}$ is the identity matrix and C is the parameter penalising training errors. Taking this approach, the C parameter can be directly included in the optimisation.

Other commonly used methodology is to rely on multiple kernel learning. The problem tackled can be then reformulated as one of convex or quadratic programming, which thus avoids local minima. KTA [11,40] and radius margin bound [18] have been approached in this way by predefining a set of kernels (which indeed can limit the performance).

Finally, other methods consider the inclusion of the optimisation stage into the original optimisation problem of SVM [39,51,54] or kernel discriminant analysis (KDA) [37].

2.5 Multi-scale case

This final subsection focuses on the multi-scale case and reviews what has been previously done in this context. As mentioned in the introduction, although the use of a multi-scale kernel can provide more appropriate representations of the feature space induced by the kernel, this usage has been barely studied in machine learning. These kernels have been mainly used with evolutionary algorithms [24, 44, 22] or gradient-based methods for specific applications [48, 10, 32]. The main problem with the evolutionary approaches is the high computational cost that is involved in the optimisation and the necessity of tuning a large number of parameters associated to the evolutionary algorithm.

With concern for the applications, in [10], an experiment of the multi-scale case with the radius margin bound is performed for handwritten digit recognition. The authors consider this experiment to be as a sanity check experiment which demonstrates the feasibility of choosing multiple kernel parameters for an SVM without leading to overfitting. This approach has been considered in the experimental part of the paper (MSRMB method). In [32], the concept of KTA (non-centred) is used to derive a method for optimising multiple hyperparameters of oligo kernels to analyse biological sequence data. Our method extends this idea by considering more robust centred KTA and general purpose Gaussian kernels and providing an extensive analysis of the potential advantages of this procedure. In [48], a gradient-based optimisation of the radius margin bound was used for the diagnosis of diffuse lung diseases. Although the performances of the SVM classifiers with spherical and multi-scale kernel in the paper do not differ significantly, the authors argue that when there is no prior knowledge about a classification problem, the multi-scale kernel should be preferred because it can be considered a more general and powerful model. A multi-scale experiment is also performed in [21]; however it achieved worse results than the spherical version at a much higher computational cost. The authors argue that this processing time increase could be due to the formulation of the optimisation problem, which requires the inversion of a matrix for each update of one of the hyperparameters. In our approach, the optimisation methodology is free of this computational requirement.

The case of multi-scale kernels is also studied in [26] where several strategies are compared. In this work, a evolutionary optimisation technique for high-dimensional search spaces based on the validation error [29] is used with the purpose of optimising a multi-scale kernel. However, the author argues that this method does not achieve a satisfactory performance and leads to overfitting in contrast to the KTA measure, which should be preferred for this purpose.

3 Multi-scale centred kernel-target alignment (MSCKTA)

This section introduces the method used in this paper to optimise all of the parameters in multi-scale kernels. The method combines the concept of centred

KTA (CKTA) with respect to the ideal kernel and a gradient ascent methodology. Furthermore, we also include a discussion of its main advantages and present a distance-based technique to initialise the gradient-ascent method.

Some attempts have been made in the literature to establish learning bounds for the Gaussian kernel with several parameters and the combination of kernels when considering large margin classifiers [40]. These studies suggest that the interaction between the margin and the complexity measure of the kernel class is multiplicative, thus discouraging the development of techniques for the optimisation of more complex and general kernels. However, recent developments have shown that this interaction is additive [53] (up to log factors), rather than multiplicative, yielding then stronger bounds. Therefore, the number of patterns needed to obtain the same estimation error with the same probability for a multi-scale kernel compared to a spherical one grows slowly (and directly depends on the pseudodimension of the kernel function, in this case the number of features). More specifically, the bound on the required sample size is $\widetilde{\mathcal{O}}(d_\phi + ||\mathbf{w}||/2)$ [53], where $\mathbf{w}$ is the SVM hyperplane and $\widetilde{\mathcal{O}}$ hides logarithmic factors in its argument, the sample size and the allowed failure probability. Note that for the spherical kernel the pseudodimension is $d_\phi = 1$ and for the multi-scale case $d_\phi = d$.

In this paper, the family of kernels is restricted to the well-known Gaussian family, which is parametrised by a d-square matrix of hyperparameters $\mathbf{Q}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{Q}(\mathbf{x}_i - \mathbf{x}_j)\right). \tag{2}$$

For the conventional Gaussian kernel (known as spherical or uni-scale), a single hyperparameter α is used (i.e., $\mathbf{Q} = \alpha^{-2}\mathbf{I}_d$, and $\mathbf{I}_d$ is the identity matrix of size d, and $\alpha > 0$), assuming that the variables are independent. However, one hyperparameter per feature (muti-scale or ellipsoidal Gaussian kernel) can also be used by setting $\mathbf{Q} = diag(\boldsymbol{\alpha}^{-2}) = diag([\alpha_1^{-2}, \ldots, \alpha_d^{-2}])$, with $\alpha_p > 0$ for all p in $\{1, \ldots, d\}$. KTA can be used to obtain the best values for α (the uni-scale method) or $\boldsymbol{\alpha}$ (the multi-scale method). Hereafter, these hyperparameters will be called kernel widths.

3.1 Ideal kernel

Although the properties of the kernel function are important, often the kernel matrix plays a more important role. Because kernel functions allow access to the feature space only via input samples, the pairwise inner products between the elements of a finite input set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are the only information that is available on the geometry of the feature space. This information is embedded in the kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where k is the kernel function. Most often, kernel algorithms work with this matrix rather than the kernel function itself. Gram matrices contain information about the similarity among the patterns; thus, the idealised kernel matrix $\mathbf{K}^*$ derived using an ideal kernel function k^*

[14] will submit the following structure:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 & \text{if } y_i = y_j \\ -1 & \text{otherwise,} \end{cases} \tag{3}$$

where y_i is the target of pattern $\mathbf{x}_i$. In other words, $\mathbf{K}^* = \mathbf{y}\mathbf{y}^{\mathrm{T}}$. $\mathbf{K}^*$ will provide information about which patterns should be considered to be similar when performing a learning task. Note that the ideal kernel can be defined only on the training patterns, in practice.

Therefore, the problem of finding an optimal set of hyperparameters $\boldsymbol{\alpha}$ is changed to the problem of finding a good approximation $\mathbf{K}_{\boldsymbol{\alpha}}$ (i.e., computed for hyperparameters $\boldsymbol{\alpha}$) for the ideal kernel matrix $\mathbf{K}^*$, given a family $\mathcal{Q}$ of kernels (see Fig.1). This way of formulating the problem allows us to separate kernel optimisation from kernel machine learning and to reduce the increase in the computational cost of learning more complex kernels (such as multi-scale ones), given that the kernel machine will be unaffected by this higher complexity.

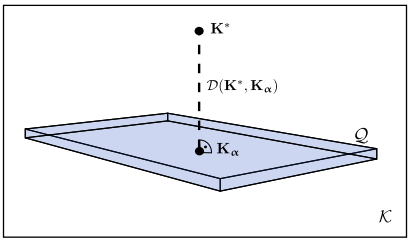


Fig. 1 The most appropriate kernel for learning is $\mathbf{K}_{\boldsymbol{\alpha}}$ (the one nearest the ideal one ($\mathbf{K}^*$) according to some measure of similarity $\mathcal{D}$, being $\mathcal{K}$ the set of positive definite kernels).

In terms of mathematical geometry, for the ideal problem presented in Fig. 1, the kernel matrix that is closest to $\mathbf{K}^*$ will be the orthogonal matrix, which can be found by maximising the angle between $\mathbf{K}_{\boldsymbol{\alpha}}$ and $\mathbf{K}^*$, an idea that is related to KTA.

The concept of ideal kernel matrix has also been reformulated to address multinomial classification [28] and regression problems [35]. Therefore, although we will focus on the binary problem, the reader can appreciate the versatility of kernel-target alignment.

3.2 Notions of kernel-target alignment (KTA) and centred KTA

Previous studies have noted several issues in KTA for different pattern distributions [14, 46]. A recent study [11] has proven a solution to this problem both empirically and theoretically using *centred kernel matrices*, a method that is based on centering the patterns in the feature space and that correlates better

with the performance than the original definition of KTA [14]. In fact, this study shows that non-centred alignment could be even negatively correlated with the accuracy in some cases, while a large positive correlation is expected of a good quality measure. However, the centred notion of alignment shows good correlation along all datasets and is always better correlated than the non-centred version.

Let us suppose an ideal kernel matrix $\mathbf{K}^*$ and a real kernel matrix $\mathbf{K}_{\boldsymbol{\alpha}}$ computed for some kernel parameters $\boldsymbol{\alpha}$. The Frobenius inner product between them ($\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^* \rangle_{\mathrm{F}} = \sum_{i,j=1}^{N} k(\mathbf{x}_i, \mathbf{x}_j) \cdot k^*(\mathbf{x}_i, \mathbf{x}_j)$, where N is the number of patterns) provides information about how *'well'* the patterns are classified in their category. Indeed, in this case, the product could be rewritten as the following equation (see Eq. (3)):

$$\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^* \rangle_{\mathrm{F}} = \sum_{y_i = y_j} k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j), \tag{4}$$

where $\sum_{y_i=y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ is related to the within-class distance, and $\sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ to the between-class distance.

The notion of centred alignment between $\mathbf{K}_{\boldsymbol{\alpha}}$ and $\mathbf{K}^*$ [14,11] is defined as:

$$\mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) = \frac{\langle \mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}, \mathbf{K}^*_{\mathrm{c}} \rangle_{\mathrm{F}}}{\sqrt{\langle \mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}, \mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}} \rangle_{\mathrm{F}} \langle \mathbf{K}^*_{\mathrm{c}}, \mathbf{K}^*_{\mathrm{c}} \rangle_{\mathrm{F}}}}, \tag{5}$$

and this quantity is totally maximised when a kernel can reflect the discriminant properties of the dataset that are used to define the ideal kernel (i.e., $\beta \mathbf{K}_{\boldsymbol{\alpha}} = \mathbf{K}^*$, where β is a scalar). $\mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) \geq 0$ because the Frobenius product of any two centred positive semi-definite matrices $\mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}$ and $\mathbf{K}^*_{\mathrm{c}}$ is non-negative.

Any kernel matrix $\mathbf{K}$ is centred by subtracting from it its empirical expectation:

$$\mathbf{K}_{\mathrm{c}} = (\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}})^{\top}(\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}}) = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{N}} - \mathbf{1}_{\frac{1}{N}}\mathbf{K} + \mathbf{1}_{\frac{1}{N}}\mathbf{K}\mathbf{1}_{\frac{1}{N}}, \tag{6}$$

where $\mathbf{Z} = \left[\Phi(\mathbf{x}_1) \cdots \Phi(\mathbf{x}_n)\right]$, $\Phi(\cdot)$ is the mapping from the input space to the feature space, and $\mathbf{1}_{\frac{1}{N}}$ is a matrix with all elements equal to $\frac{1}{N}$. $\mathbf{K}_{\mathrm{c}}$ will also be a positive semi-definite kernel matrix that satisfies $k(\mathbf{x}, \mathbf{x}) \geq 0, \forall\ \mathbf{x} \in \mathcal{X}$ and symmetry.

The concentration bound for CKTA and the proof that there exists good alignment-based predictors both for regression and classification can be seen in [11].

3.3 Optimisation of MSCKTA

Because of the differentiability of $\mathcal{A}_{\mathrm{c}}$ with respect to the kernel width vector $\boldsymbol{\alpha}$, a gradient ascent algorithm can be used to maximise the alignment between the kernel that is constructed using $\boldsymbol{\alpha}$ and the ideal kernel, as follows:

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*). \tag{7}$$

Because $\boldsymbol{\alpha}$ is a vector that is composed from several variables, we will have a gradient vector that is composed of partial derivatives $\nabla \mathcal{A}_{\mathrm{c}} = \left[\frac{\partial \mathcal{A}_{\mathrm{c}}}{\partial \alpha_1}, \ldots, \frac{\partial \mathcal{A}_{\mathrm{c}}}{\partial \alpha_d}\right]$, where d is the data dimensionality. In this work, the iRprop$^+$ algorithm is used to optimise the aforementioned centred KTA, because of its proven robustness [33]. Each parameter α_i will be updated considering the sign of $\frac{\partial \mathcal{A}_{\mathrm{c}}}{\partial \alpha_i}$ but not the magnitude. Although the second partial derivatives can also be computed and used for optimisation, they could actually make this process more computationally costly due to the complexity of this second derivative formula. The alignment derivative with respect to the kernel widths $\boldsymbol{\alpha}$ (see Eq. (5)) is:

$$\frac{\partial \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*)}{\partial \boldsymbol{\alpha}} = \frac{1}{||\mathbf{K}^*_{\mathrm{c}}||_{\mathrm{F}}} \left[\frac{\left\langle \frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}, \mathbf{K}^*_{\mathrm{c}} \right\rangle_{\mathrm{F}}}{||\mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}||_{\mathrm{F}}} - \frac{\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*_{\mathrm{c}} \rangle_{\mathrm{F}} \cdot \left\langle \mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}, \frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}} \right\rangle_{\mathrm{F}}}{||\mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}||^3_{\mathrm{F}}} \right], \quad (8)$$

where, $||\mathbf{A}||_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_{\mathrm{F}}}$ and for arbitrary matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, it is satisfied that $\langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_1, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_2 \rangle_{\mathrm{F}}$ [11], which simplifies the computation. Note that the derivative for α_i is computed taking into account the other kernel parameters $\alpha_{j|j\neq i}$ because $\mathbf{K}_{\boldsymbol{\alpha}}$ is included in the formulation. The computation of the KTA takes $\mathcal{O}(N^2)$ operations per parameter α to optimise [26]. Because this optimisation does not involve any additional optimisation problem it is very fast in practice. Therefore, the computational complexity of MSCKTA is moderated.

For the spherical Gaussian kernel, $\boldsymbol{\alpha} = \alpha \cdot \mathbf{1}$ and the derivative with respect to α can be computed as

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \alpha} \right) = \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\alpha^3} \cdot \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2} \right). \quad (9)$$

However, for the case of the multi-scale Gaussian kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{z=1}^{d} \frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2} \right) = \prod_{z=1}^{d} \exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2} \right), \quad (10)$$

the derivative is the following:

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \alpha_h} \right) = \frac{(x_{ih} - x_{jh})^2}{\alpha_h^3} \cdot \prod_{z=1}^{d} \exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2} \right). \quad (11)$$

The specific details and pseudo-code of the iRProp+ algorithm can be checked in [33]. To avoid including positivity constraints in the optimisation problem of $\boldsymbol{\alpha}$ (note that α should vary from 0 to $+\infty$), a logarithmic scale (base 10) is used for the parametrisation, which does indeed result in a more stable optimisation. In other words, we consider $\boldsymbol{\alpha} = \{10^{\alpha'_1}, \ldots, 10^{\alpha'_d}\}$ and optimise the functional with respect to $\boldsymbol{\alpha}' = \{\alpha'_1, \ldots, \alpha'_d\}$, avoiding the inclusion of any constraint for $\boldsymbol{\alpha}'$.

For the case of multiple parameters, in which independent widths are adjusted for each feature, one could think that maximising KTA too much could

lead to over-fitting. However, the definition of KTA avoids this type of behaviour with the first term in the Frobenius product (see Eq. (4)), which prevents the kernel from a width that is too narrow (which is known to result in overfitting as we will see). To clarify this in the usefulness of centred KTA, consider the following potential matrices derived from a Gaussian kernel for a dataset comprised of four patterns ($\mathbf{x}_1 \in \mathcal{C}_1$ and $\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \in \mathcal{C}_2$):

$$\mathbf{K}^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \ \mathbf{K}_{\alpha\to 0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$
$$\mathbf{K}_{\alpha\to\infty} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

where $\mathbf{K}^*$ represents the ideal matrix, i.e., the matrix that perfectly maintains the labelling structure. $\mathbf{K}_{\alpha\to 0}$ and $\mathbf{K}_{\alpha\to\infty}$ are both considered to be trivial solutions that are to be avoided when using kernel machines. More specifically, with $\mathbf{K}_{\alpha\to 0}$ each pattern is only similar to itself (i.e., the result of choosing a too narrow kernel width):

$$\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2} \gg 0, \ \forall \mathbf{x}_i \neq \mathbf{x}_j, \tag{12}$$

a solution that, in practice, leads to overfitting. With $\mathbf{K}_{\alpha\to\infty}$, however, our kernel matrix does not incorporate any information about the domain because all of the patterns are considered to be similar:

$$\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2} \simeq 0, \ \forall \mathbf{x}_i, \mathbf{x}_j, \tag{13}$$

With regard to non-centred KTA, it is noticeable that the classes in the previous example are unbalanced in a relationship of 3 to 1. Therefore, the number of $+1$ terms in the ideal matrix is higher than the number of -1 terms. This fact, which could appear to be insignificant, is indeed blameworthy for the KTA pattern distribution issues because the constructed kernel is closer to the $\mathbf{K}_{\alpha\to\infty}$ matrix when this imbalance ratio is higher. Because CKTA takes into account the centred kernel matrix (by subtracting from each point the mean of the corresponding column and row and the mean of the complete matrix), this approach is free from this problem. To see this, analyse the difference between Eq. (4) (which is used for KTA) and the following equation (used for centred KTA):

$$\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} = \sum_{y_i = y_j} (1 - m_{r_i} - m_{c_j} + m) \cdot k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{y_i \neq y_j} (1 + m_{r_i} + m_{c_j} - m) \cdot k(\mathbf{x}_i, \mathbf{x}_j), \tag{14}$$

where $\mathbf{K}^* = \mathbf{y}\mathbf{y}^{\mathrm{T}}$, $[\mathbf{K}_{\mathrm{c}}^*]_{ij} = y_i y_j - m_{r_i} - m_{c_j} + m$, $m_{r_i} = \frac{1}{N}\sum_{z=1}^{N} y_i y_z$, $m_{c_j} = \frac{1}{N}\sum_{z=1}^{N} y_j y_z$ and $m = \frac{1}{N^2}\sum_{z,h=1}^{N} y_z y_h$. Note that both $(1 - m_{r_i} - m_{c_j} + m)$ and $(1 + m_{r_i} + m_{c_j} - m)$ are dependent on the label distribution and are used as a weighting procedure.

The results obtained for KTA and CKTA in an imbalanced toy dataset are shown in Fig. 2. In this case, it can be seen that the optimal kernel parameter (α value with maximum alignment) for KTA and CKTA are different: around 10^2 for KTA and 10^{-2} for CKTA. Furthermore, in the bottom part of the figure, where the two solutions are plotted, it can be seen that the kernel value obtained for CKTA is more appropriate for the discrimination of the classes (KTA tends to choose solutions that consider that all the patterns are similar to the rest by setting $\alpha \to \infty$).

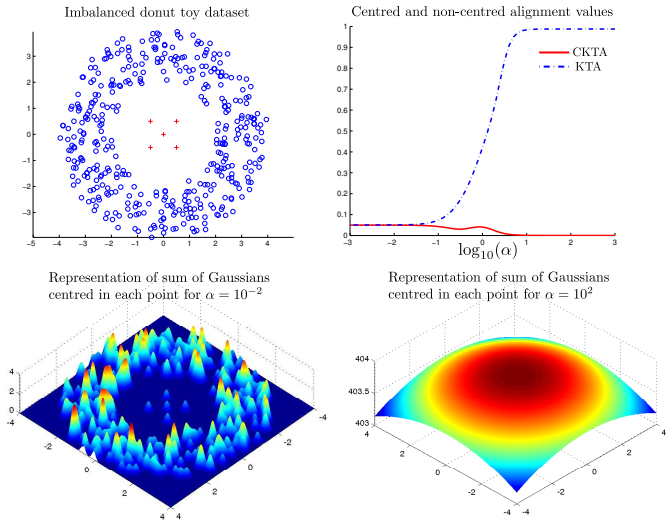


Fig. 2 Two-dimensional imbalanced toy dataset and alignment values obtained for different α values (considering CKTA and KTA).

Finally, Fig. 3 shows two toy datasets and the corresponding alignment optimisation surface, where it can be appreciated the necessity of the use of a multi-scale kernel. As can be seen, the optimum values are located in regions where $\alpha_1 \neq \alpha_2$.

3.4 Initialisation scheme of the Gaussian kernel parameters

From the KTA definition it follows that patterns belonging to the same class should present a high similarity (in terms of a distance relation), in contrast to patterns belonging to different classes. This idea could be exploited to obtain an inital value of the parameters $\boldsymbol{\alpha}$ of the Gaussian kernel by fitting a probability distribution to the set containing the within-class distances $\mathbf{d}_w$ (e.g. a set containing the distance of each pattern to all of the patterns belonging to the same class), in such a way that the lower the distance, the greater the similarity and the probability of belonging to the same class. As a possible way to

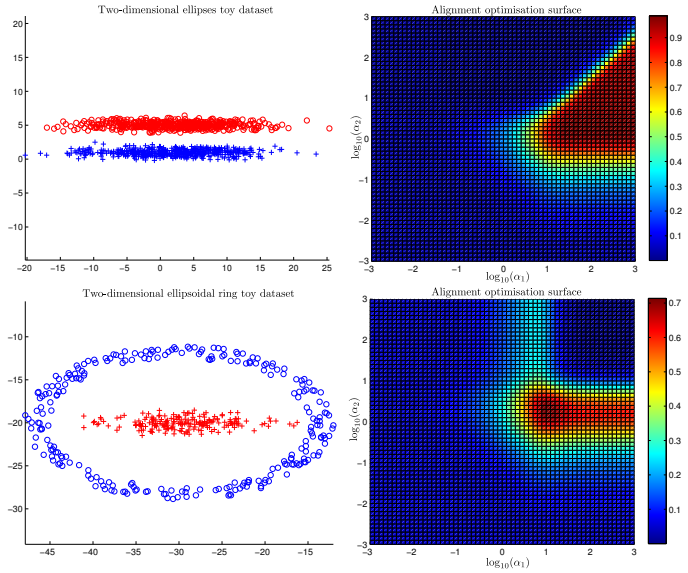


Fig. 3 Two-dimensional toy datasets presenting different class variances per feature and their alignment values when using a grid of values for α_1 and α_2.

do this, we can assume the exponential distribution $f(\mathbf{d}_w, \lambda) = \lambda \exp(-\lambda \mathbf{d}_w)$, where a close relation can be found between the λ parameter of this exponential distribution and the α parameter in the Gaussian kernel (considering now one single parameter for the kernel). The connection can be seen by analysing the following equation and comparing it to the exponential distribution:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\lambda \cdot ||\mathbf{x}_i - \mathbf{x}_j||^2\right), \quad \lambda = \frac{1}{2\alpha^2}, \quad ||\mathbf{x}_i - \mathbf{x}_j||^2 \in \mathbf{d}_w \text{ if } y_i = y_j. \tag{15}$$

Note that the first multiplier in the exponential distribution (i.e. λ) is not required, since the Gaussian kernel does not need that its integral sum to one. However, it is more realistic for real world problems to assume a local neighbourhood-based similarity notion (e.g. for nonlinearly separable problems or multimodal ones), considering that each pattern should be similar to their k-nearest neighbours belonging to the same class. Then, denote $\mathbf{d}_w = \{\mathbf{d}_{w+}, \mathbf{d}_{w-}\}$, where:

$$d_{w+}^{ij} = ||\mathbf{x}_i - \mathbf{x}_j||^2, \quad y_i, y_j = +1 \tag{16}$$

where $\mathbf{x}_j$ is one of the k-nearest neighbours of $\mathbf{x}_i$ ($k = 5$ is selected for simplicity). The analogous equation is used for the negative class. Note that, for the exponential distribution, the parameter λ is estimated as the mean of $\mathbf{d}_w$. Then, the kernel parameter can be determined as $\alpha = \sqrt{\lambda/2}$. For the multiscale case, the input features are assumed to be independent, in such a way that λ_i (and therefore α_i) for feature i are computed independently (considering the distance of the patterns for that feature). The result obtained by

means of this procedure for the ellipsoidal ring dataset in Fig. 3 can be seen in Fig. 4 where $\alpha_1 = 10^{-0.53}$ and $\alpha_2 = 10^{-0.87}$ (different values per feature).

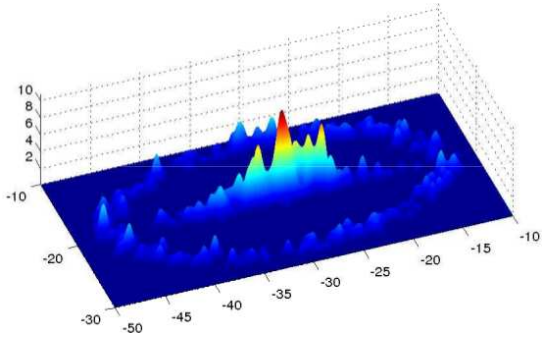


Fig. 4 Representation of the sum of multi-scale Gaussians centred in each point for the elliposoidal ring toy dataset. The optimal parameter values obtained by the proposed initialisation scheme are $\alpha_1 = 10^{-0.53}$ and $\alpha_2 = 10^{-0.87}$ (note that the data has been standardised beforehand).

The main intuition behind this technique is that the kernel parameters are selected depending on the data itself, i.e. on the distance to close patterns belonging to the same class, in order to construct local neighbourhoods of similar patterns. Up to the authors knowledge, this is the first attempt to propose a deterministic method for initialising the Gaussian kernel parameters in this context.

3.5 Filtering non-informative features for the construction of the kernel matrix

An important characteristic of multi-scale kernels which we would like to highlight is that they provide the opportunity to perform feature selection by filtering attributes with large α_z values. When the Gaussian kernel width $\alpha_z \to \infty$, the kernel matrix computed for that unique feature remains invariant and tends to a matrix of ones, which can be interpreted as feature z not being used for the kernel computation (see Eq. (10)), an omission that could be beneficial for model interpretability. Note that if this does not occur, the feature will include noise in the kernel computation. In this subsection, we show that if feature z is non-informative, $\alpha_z \to \infty$ will be considered as an optimum value for the gradient ascent algorithm.

Consider the case of a variable of index z that, for all value of the parameter α_z for the kernel, it fulfills that:

$$\left(\frac{N_{y_i}}{N}|\{(x_{iz} - x_{jz})^2 \leq 2\alpha_z^2\}|_{y_i=y_j}\right) = \left(\frac{N_{y_j}}{N}|\{(x_{iz} - x_{jz})^2 \leq 2\alpha_z^2\}|_{y_i\neq y_j}\right), \tag{17}$$

where $|\{\cdot\}|$ denotes the cardinality of the set, and N_{y_i} the number of patterns with label equal to y_i. Therefore, this variable can be said to be noisy for all value chosen for the width of the Gaussian for x_{iz} (i.e., the notion of similarity does not report information for the classification problem). If this holds for variable z, then:

$$\left(\sum_{y_i=y_j}(1-m_{r_i}-m_{c_j}+m)\cdot k(x_{iz},x_{jz})\right)\simeq \left(\sum_{y_i\neq y_j}(1+m_{r_i}+m_{c_j}-m)\cdot k(x_{iz},x_{jz})\right). \tag{18}$$

Under this assumption, for variable z, it holds that

$$\langle \mathbf{K}_{\alpha_z}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} \simeq 0,\ \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\alpha_z}, \mathbf{K}^*) \simeq 0. \tag{19}$$

where $\mathbf{K}_{\alpha_z}$ is a kernel matrix where the only variable used for distance computation is the variable z.

Recall that for the multi-scale case $\mathbf{K}_{\boldsymbol{\alpha}} = \mathbf{K}_{\alpha_1} \circ \ldots \circ \mathbf{K}_{\alpha_d}$. Now, consider that the set of features except feature z perfectly represent the output space, i.e.:

$$\beta(\mathbf{K}_{\alpha_1} \circ \ldots \circ \mathbf{K}_{\alpha_{z-1}} \circ \mathbf{K}_{\alpha_{z+1}} \circ \ldots \circ \mathbf{K}_{\alpha_d}) = \mathbf{K}^*, \tag{20}$$

where β is a scalar and $\circ$ represents the hadamard or entrywise product between matrices, i.e., for two matrices A and B of the same dimension, the hadamard product $(A \circ B)$ is another matrix (of the same dimension) with elements given by: $(A \circ B)_{i,j} = (A)_{i,j} \cdot (B)_{i,j}$.

In this way, the complete kernel matrix can be decomposed as $\mathbf{K}_{\boldsymbol{\alpha}} = \mathbf{K}^* \circ \mathbf{K}_{\alpha_z}$ with the informative features in $\mathbf{K}^*$ and the non-informative one in $\mathbf{K}_{\alpha_z}$. To analyse how the non-informative variable of index z interferes in the kernel matrix, note that:

$$\langle \mathbf{K}_{\alpha_z}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} < \langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} \leq \langle \mathbf{K}^*, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}}, \tag{21}$$

because $\langle \mathbf{K}_{\alpha_z}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} \simeq 0$ and the addition of a non-informative feature will never decrease the angle of the matrix with respect to the ideal one. Given that the maximum alignment is $\mathcal{A}_{\mathrm{c}}(\mathbf{K}^*, \mathbf{K}^*) = 1$ and we know that $\mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) \leq \mathcal{A}_{\mathrm{c}}(\mathbf{K}^*, \mathbf{K}^*)$, the gradient of the alignment will converge to the best solution $\mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) = \mathcal{A}_{\mathrm{c}}(\mathbf{K}^*, \mathbf{K}^*) = 1$, which is true for (see Eq. (5)):

$$\langle (\mathbf{K}^* \circ \mathbf{K}_{\alpha_z})_{\mathrm{c}}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} = \sqrt{\langle (\mathbf{K}^* \circ \mathbf{K}_{\alpha_z})_{\mathrm{c}}, (\mathbf{K}^* \circ \mathbf{K}_{\alpha_z})_{\mathrm{c}} \rangle_{\mathrm{F}} \langle \mathbf{K}_{\mathrm{c}}^*, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}}} \tag{22}$$

$$\mathrm{Tr}((\mathbf{K}^* \circ \mathbf{K}_{\alpha_z})_{\mathrm{c}} \cdot \mathbf{K}_{\mathrm{c}}^*)^2 = \mathrm{Tr}((\mathbf{K}^* \circ \mathbf{K}_{\alpha_z})_{\mathrm{c}}^2) \cdot \mathrm{Tr}((\mathbf{K}_{\mathrm{c}}^*)^2), \tag{23}$$

where $\mathrm{Tr}(\mathbf{A})$ corresponds to the trace of $\mathbf{A}$. The only case that fulfils this is $\mathbf{K}_{\alpha_z} = \mathbf{1}$, and this is the case when $\alpha_z \to \infty$ (see Section 3.3), because all the patterns are considered to be equally similar. Therefore, from Eq. (11), $\frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \alpha_z} \to 0$ and $\frac{\partial \mathcal{A}_{\mathrm{c}}}{\partial \alpha_z} \to 0$. Consequently, as the derivative is equal to zero, the

case of $\alpha_z \to \infty$ will be an optimum for the gradient-based optimisation algorithm. Note that this filtering is done implicitly without including any sparsity coefficient in the optimisation. Therefore, only non-informative features are removed. However, as it is well-known, whether the gradient ascent algorithm reaches the optimum point depends on the initialisation itself.

The remaining methods studied in this paper do not naturally perform any type of feature selection (i.e., a sparsity coefficient could be added to the optimisation but this step is not performed explicitly) because adding non-informative dimensions to the problem should not damage the SVM solution. This is due to the fact that the capacity control performed by the SVM method is equivalent to some form of regularisation so that "denoising" is not necessary [49]. In the case of KTA, the optimisation performed recognises directly the variables that do not report information about the labelling or that are very noisy. KTA applied for the purpose of deciding most informative variables (i.e., to perform feature selection) has been only investigated in [46] where KTA is used to optimise a weighting variable for each feature by a gradient ascent algorithm.

4 Experimental results

This section aims to provide an extensive empirical analysis of the use of multi-scale kernels. Firstly, the goodness of this type of kernel is analysed by plotting an approximation of the feature space that is induced by the kernel. Secondly, several approaches to uni and multi-scale kernels are tested for comparison purposes for a set of 24 binary benchmark datasets, and statistical tests are conducted to analyse whether the method previously presented improves their performance significantly. Thirdly, the feature selection performed by the methodology is analysed and a deeper analysis of the situations in which a multi-scale approach is useful is done. Finally, an analysis of the results for different initialisations is presented.

Regarding the experimental setup, a stratified 10-fold cross-validation was applied to divide the data, using the same partitions for all of the methods that were compared. Because the SVM solution is unique, one model was obtained and evaluated for each split. The results are taken as the mean and standard deviation over each of the 10 test sets.

As stated before, the optimisation of the gradient-based methods is guaranteed only to find a local minimum; therefore, the quality of the solution can be sensitive to initialisation. Two different approaches are considered in this case. For the comparison with other state-of-the-art methods, the initial point for all of the methods tested was fixed at 10^0 (as suggested by other studies [10]). As a different part of the experimental study, we also compare this fixed choice (10^0) with random initialisation and with the deterministic initialisation technique proposed in subsection 3.4. The gradient norm stopping criterion was set at 10^{-5} and the maximum number of conjugate gradient steps at 10^2 [33].

Table 1 Characteristics for the 24 datasets tested, ordered by the number of attributes d.

Dataset	N	d	Dataset	N	d
haberman (HA)	306	3	hepatitis (HE)	155	19
listeria (LI)	539	4	bands (BA)	365	19
mammographic (MA)	830	5	heart-c (HC)	302	22
monk-2 (MO)	432	6	labor (LA)	57	29
appendicitis (AP)	106	7	sick (SI)	3772	33
pima (PI)	768	8	krvskp (KR)	3196	38
glassG2 (GL)	163	9	credit-a (CR)	690	43
saheart (SA)	462	9	specftheart (SP)	267	44
breast-w (BW)	699	9	card (CA)	690	51
heartY (HY)	270	13	sonar (SO)	156	60
breast (BR)	286	15	colic (CO)	368	60
housevotes (HO)	232	16	credit-g (CG)	1000	61

All nominal variables are transformed into binary ones

Several benchmark binary datasets that have different characteristics were tested. Table 1 shows the characteristics of these datasets, where the number of patterns (p) and attributes (a) can be observed. These publicly available real classification datasets were extracted from the UCI repository [2].

4.1 Graphical comparisons

This subsection explores the notion of empirical feature space to analyse the behaviour of a multi-scale kernel via a graphical experiment. The empirical feature space can be defined as a Euclidean space that preserves the dot product information about $\mathcal{H}$ that is contained in $\mathbf{K}$ (i.e., this space is isomorphic to the embedded feature space $\mathcal{H}$, but it is Euclidean). It is possible to verify that the kernel matrix of the training images that are obtained by this transformation corresponds to $\mathbf{K}$, when considering the standard dot product [47, 61]. This methodology provides us with the opportunity to limit the dimensionality of the space by computing the eigendecomposition of $\mathbf{K}$ and choosing the r dominant eigenvalues (and their associated eigenvectors) to project the data while approximating the structure of $\mathcal{H}$. That is, for example, the best rank-2 approximation to $\mathbf{K}$ is $\hat{\mathbf{K}} = \sum_{i=1}^{2} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$, in the sense that minimises $||\mathbf{K} - \hat{\mathbf{K}}||_{\mathrm{F}}^2$ over all rank-2 matrices (where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm, λ_i corresponds to the i-th highest eigenvalue and $\mathbf{u}_i$ the eigenvector associated to eigenvalue λ_i).

Therefore, an approximation of the feature space can be plotted by means of the first most-representative dimensions of this empirical feature space [61]. We use this method to represent the embedding space induced by CKTA and MSCKTA optimisation for several datasets (see Fig. 5). It can be appreciated from Fig. 5 that when a multi-scale kernel is used (right plot of each dataset), the space induced appears to be better determined because the class separability is clearer (thus leading to simpler decision functions). Furthermore, Fig. 5 includes information of the eigenvalues of both matrices (i.e., the matrix induced by CKTA and the matrix induced by MSCKTA). This information is represented by a γ value, that corresponds to $\frac{\mu_1+\mu_2}{\sum_{i=1}^{N}\mu_i}$, where μ_i is the i-th

eigenvalue for a given matrix ordered in descending order. From these values, it can be observed that the normalised sum of the first eigenvalues is higher for the kernel matrix computed by MSCKTA, indicating this that these two dimensions do incorporate more information about the kernel matrix we are diagonalising. Indeed, previous studies in the literature [8] have demonstrated that when a kernel presented a higher normalised sum of the first eigenvalues (applying kernel principal component analysis) than other kernel function it is because the first kernel suited the underlying problem better. In our case, because we are not applying kernel principal components analysis, but a reduction of the empirical kernel map instead, this γ value does not represent the total of data variance covered, but rather the total information represented of the original kernel matrix.

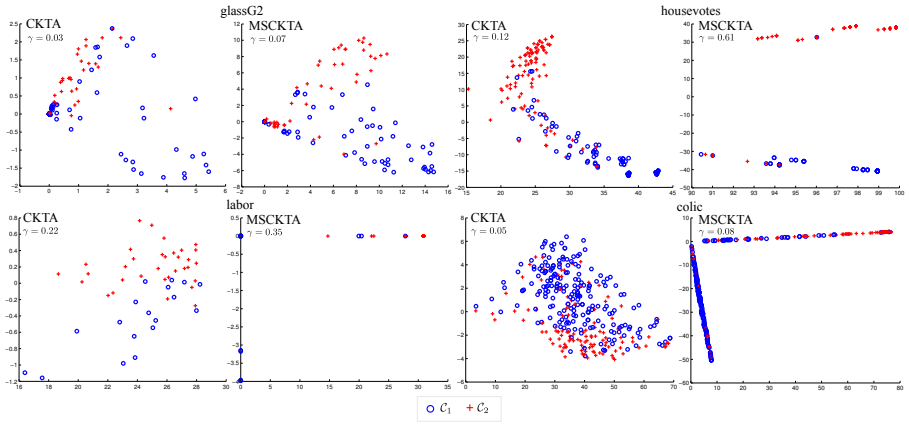


Fig. 5 Graphic showing the 2-dimensional approximation of the empirical feature space induced by CKTA optimisation and the uni-parameter kernel (left plot for each dataset) and by MSCKTA optimisation and the multi-scale kernel (right plot for each dataset).

4.2 Comparisons to other methodologies

The following methods were compared in the experimentation because they can be considered to be very representative methods in kernel optimisation:

- Cross-validation (CV) using a stratified nested 5-fold cross-validation on the training sets with a single kernel parameter and the C parameter of SVM selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.
- Centred kernel-target alignment for optimisating a convex combination of kernels through multiple kernel learning (AMKL) [11]. The kernels used for the optimisation are the ones associated to the different kernel width values (i.e., $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$). Once the kernel width is adjusted, the regularisation parameter C of SVM is tuned by minimising the classification error

estimated by a stratified nested 5-fold cross-validation on the training sets (with the parameter C within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$). This two stage optimisation method is also referred in the literature as second-order method [9].
- Smoothed span of support vectors (SSV) optimised using a gradient-based methodology [10]. A spherical kernel is used and the optimisation of C is made together with the kernel parameter (considering Eq. (1)).
- Evidence maximisation (EVID) and its multi-scale version (MSEVID), optimised through a gradient-based methodology [52]. The optimisation of C is made considering Eq. (1).
- Smoothed radius margin bound (RMB) and its multi-scale version (MSRMB), optimised using a gradient-based methodology [10]. The optimisation of C is made considering Eq. (1).
- Centred kernel-target alignment (CKTA) and multi-scale centred kernel-target alignment (MSCKTA), optimised using a gradient ascent methodology. Once the kernel width is adjusted, the regularisation parameter C of SVM is tuned by minimising the classification error estimated by a stratified nested 5-fold cross-validation on the training sets (with the parameter C within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$), as in other studies [32].

Each benchmark dataset was appropriately standardised before the learning process. As suggested in [10], for SSV, EVID, MSEVID, RMB and MSRMB, a modified version of the Polack-Ribiere flavour of conjugate gradients was used to compute the search directions; a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria were used together with the slope ratio method to determine the initial step sizes. Besides, the first and second derivatives were used for the optimisation. For CKTA and MSCKTA, the iRprop^{+} optimisation method [33] has been selected because of its good behaviour in alignment optimisation [32] only using the first derivative due to the complexity of the second derivative formula, which could make the whole process more computationally costly.

The source codes in Matlab for CV, AMKL, CKTA and MSCKTA are available, together with all of the datasets, partitions and results (in terms of mean and standard deviation for all of the datasets, methods and metrics considered) on the website associated with this paper[2]. SSV, EVID, MSEVID, RMB and MSRMB have been tested by using the Matlab code[3] provided by Chapelle.

All of the algorithms were tested with the $L2$ Support Vector Classification (SVC) paradigm (in order to fairly compare with [10]).

Table 2 shows the test mean rankings (1 for the best method and 9 for the worst) and the mean test performance along all of the 24 datasets in terms of the accuracy (*Acc*), the number of support vectors (SVs), and the centred alignment for training ($\mathcal{A}_{tr}$) and testing sets ($\mathcal{A}_{ts}$). The number of support vectors has been reported because it was noticed that the value chosen for the

[2] http://www.uco.es/grupos/ayrna/gbmskta
[3] http://olivier.chapelle.cc/ams/

cost parameter C decreases when kernel-target alignment was used. This cost parameter controls the trade-off between allowing training errors and forcing rigid margins, in such a way that when $C \to \infty$ the SVM leads to the hard-margin approach. Therefore, if C is too large, we would have a high penalty for non-separable points and could store too many support vectors, which could lead to overfitting.

Table 2 Mean test values and rankings obtained for all the methods tested and the following metrics: Accuracy (Acc), number of support vectors (SVs), training alignment ($\mathcal{A}_{tr}$) and testing alignment ($\mathcal{A}_{ts}$).

Methodology	CV	AMKL	SSV	EVID	MSEVID	RMB	MSRMB	CKTA	MSCKTA
Average Acc	*84.33*	84.21	81.11	80.04	76.62	79.92	77.15	82.28	**85.21**
Average ranking	4.19	4.56	5.69	5.75	6.64	5.50	6.37	*3.94*	**2.35**
Average SVs	*292.15*	379.71	370.01	407.27	489.63	417.68	498.34	292.61	**289.62**
Average ranking	**2.08**	5.12	3.98	5.56	6.41	6.35	7.48	4.18	*3.81*
Average $\mathcal{A}_{tr}$	0.221	*0.231*	0.184	0.195	0.138	0.201	0.151	0.227	**0.379**
Average ranking	5.29	3.33	6.56	6.92	6.79	5.89	5.91	*3.29*	**1.00**
Average $\mathcal{A}_{ts}$	0.211	*0.218*	0.161	0.175	0.110	0.180	0.125	0.212	**0.352**
Average ranking	4.42	*3.58*	6.81	6.75	6.79	5.73	5.92	3.92	**1.08**

The best method is in **bold** face and the second one in *italics*.

From these results, several conclusions can be drawn. First, the good performance of the MSCKTA method can be observed by analysing the mean Acc ranking, since it outperforms the other methods, especially the other multi-scale approaches (i.e., MSEVID and MSRMB). Indeed, all of the methods based on kernel-target alignment (i.e., AMKL, CKTA and MSCKTA) appear to achieve acceptable results when compared to the rest of estimators. Specifically, the goodness of the gradient ascent methodology can be observed when using the multi-scale version. The poor performance of the other multi-scale approaches (compared to the uni-scale versions) could be for to two different reasons: first, the difficulty of optimising the parameters in such a high-dimensional search space (because there could be more directions to move to undesired local optima [26]), and second, the nature of the estimator because, for example, SSV and RMB are considered to be loose bounds on the generalisation errors (this problem has been noted in the literature [19]).

Furthermore, despite the use of a more complex kernel, it can be noted that the models obtained using MSCKTA are simpler (i.e., sparser models in terms of the number of support vectors) than the models obtained using the other kernel optimisation methods. This simplicity could result from using a more complex map, which therefore leads to a more "ideal" transformation of the input space, using the term ideal in the sense of the kernel mapping leading to a perfectly linearly separable set in the feature space.

Finally, when analysing the alignment results ($\mathcal{A}_{tr}$ and $\mathcal{A}_{ts}$), several statements in the literature can be validated. First, the use of the multi-scale approach leads to a far better alignment. Indeed, using this type of kernel achieves even better alignment values than a combination of kernels (AMKL). Second, the training ($\mathcal{A}_{tr}$) and testing alignment ($\mathcal{A}_{ts}$) appear to be highly correlated

such that the high alignment values in the training set could be robust enough for determining the optimal kernel width without any loss of generality. More specifically, the estimate of the alignment can be said to be concentrated (i.e., the probability of deviation from the mean decays exponentially), meaning that when a high alignment is obtained on the training set, a high alignment is expected in the testing. Last, but not least, similar alignment values were reported for CV (0.221 and 0.211) and CKTA (0.227 and 0.212), which shows the relationship between alignment optimisation (CKTA) and accuracy optimisation (CV).

Although the necessity of using an ellipsoidal or multi-scale kernel is inherent to the nature of the features of the problem, the probability that the dataset presents attributes that have very different scales is higher as the number of features grows. This hypothesis can be observed in Fig. 6, where the mean accuracies for each dataset are represented for CKTA and MSCKTA and the datasets have been ordered according to the number of features. As observed, when the number of features is high the differences between the methodologies grow and the importance of using multiple hyperparameters is thus demonstrated.

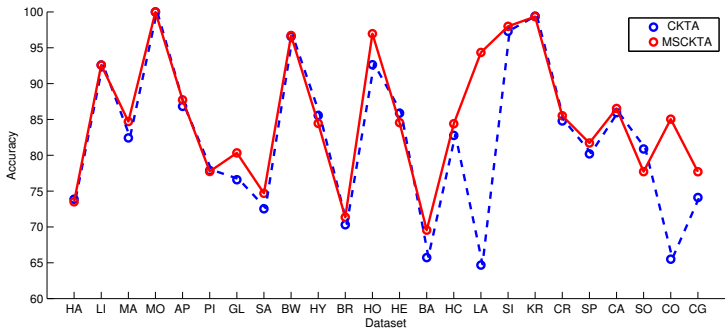


Fig. 6 Accuracy values for CKTA and MSCKTA.

To analyse the value of the results, the non-parametric Friedman's test [16] (with $\alpha = 0.05$) has been applied to the mean *Acc* rankings, rejecting the null-hypothesis that all of the algorithms perform similarly in mean. The confidence interval was $C_0 = (0, F_{(\alpha=0.05)} = 1.99)$, and the corresponding F-value was $7.28 \notin C_0$. The Holm test for multiple comparisons was also applied (see Table 3), and the test concluded that there were statistically significant differences in mean *Acc* ranking for $\alpha = 0.05$ when the MSCKTA was selected as the control method for all of the methods considered.

Table 4 includes the mean runtime values used to optimise all of the parameters for the SVM method for all of the optimisation methods considered. This time includes the seconds needed to adjust all the hyperparameters (by cross-validation or by gradient-descent depending on the parameter and the method), but not the time needed for training and testing the model after-

Table 3 Comparison in mean *Acc* ranking of the different algorithms using the Holm procedure with MSCKTA as the control algorithm.

i	Algorithm	z	p-value	Adjusted alpha
1	MSEVID	5.42857	0.00000_*	0.00625
2	MSRMB	5.08599	0.00000_*	0.00714
3	EVID	4.29542	0.00002_*	0.00833
4	SSV	4.21637	0.00002_*	0.01000
5	RMB	3.97920	0.00007_*	0.01250
6	AMKL	2.79334	0.00522_*	0.01667
7	CV	2.31900	0.02040_*	0.02500
8	CKTA	2.00277	0.04520_*	0.05000

$_*$: statistically significant differences for $\alpha = 0.05$.

wards. It can be seen that the methods based on CKTA optimising a spherical kernel are computationally efficient (AMKL and CKTA) and present a computational complexity similar to CV, resulting then in a suitable optimisation technique for kernel learning purposes. Furthermore, MSCKTA also obtains reasonable time results (note for example the case of the sonar dataset where there were 60 parameters to optimise but only took 142 secs because of the low number of patterns). Observe that, from all the multi-scale methods, MSCKTA reports an average computational time. The computational time for MSRMB is lower but at the cost of serious performance degradation (see Table 2).

Table 4 Mean runtime values (sec) to optimise the parameters with the different methods considered.

Dataset	CV	AMKL	SSV	EVID	MSEVID	RMB	MSRMB	CKTA	MSCKTA
haberman	11.05	*6.30*	10.48	27.42	30.89	6.55	8.91	**5.84**	11.82
listeria	*20.28*	24.55	49.06	141.60	171.55	42.89	36.01	**6.86**	71.12
mammographic	48.83	*42.14*	127.54	392.15	287.56	82.35	126.17	**30.19**	108.69
monk-2	*16.54*	17.82	36.88	86.12	139.76	38.50	28.40	**6.49**	25.04
appendicitis	6.52	*2.45*	6.31	7.93	7.54	3.87	3.41	**1.54**	9.21
pima	*40.45*	48.22	234.89	120.15	220.94	81.76	217.29	**25.56**	132.94
glassG2	8.20	3.19	12.81	11.05	24.78	*2.79*	18.26	**2.28**	15.29
saheart	*19.02*	24.00	66.14	27.39	51.17	21.54	49.92	**11.38**	122.60
breast-w	32.86	*14.79*	68.49	173.00	533.76	73.22	24.78	**14.42**	71.61
heartY	11.15	4.62	15.94	20.90	12.63	26.36	*4.59*	**3.73**	46.25
breast	12.52	*5.71*	14.37	25.35	12.08	6.52	20.67	**4.66**	39.05
housevotes	10.13	4.24	10.72	14.14	7.58	14.72	*4.07*	**2.91**	34.71
hepatitis	8.34	5.20	3.67	6.39	6.69	6.87	**1.97**	*2.38*	40.27
bands	17.05	*5.78*	25.33	44.12	21.75	25.93	11.53	**5.66**	79.81
heart-c	14.43	6.64	23.09	29.44	22.10	30.17	*5.97*	**5.01**	115.16
labor	7.06	5.09	*1.40*	1.83	1.78	**1.35**	1.54	1.89	30.87
sick	**1385.19**	1425.50	1676.19	5987.80	37008.14	2766.37	3869.13	**1180.89**	9137.40
krvskp	1172.12	**636.69**	1238.80	7801.57	11973.69	1795.19	1821.03	*1004.08*	10010.25
credit-a	63.51	*58.81*	198.65	188.06	239.42	100.87	64.42	**36.69**	569.15
spectfheart	16.16	12.75	8.42	29.47	31.15	*7.74*	12.85	**7.43**	178.93
card	70.96	60.51	209.00	99.27	233.04	*40.22*	76.28	**37.16**	802.58
sonar	10.95	12.89	11.79	*3.75*	17.36	**1.89**	7.41	3.93	142.61
colic	28.22	11.44	17.67	15.93	66.79	**6.16**	22.24	*7.54*	360.12
credit-g	163.56	172.97	177.30	140.42	710.80	31.91	199	72.08	1952.35
Mean	133.13	*108.84*	176.87	641.47	2159.71	217.32	276.49	**103.36**	1004.49

4.3 Feature selection

Not only can MSCKTA be useful in many real-world applications that present very different attributes, but it also appears to outperform uni-scale approaches (in accuracy) and obtain sparser models than some state-of-the-art methods. Moreover, as stated above, another advantage is that it provides us with the opportunity to perform feature selection by filtering attributes with large α_i values. Table 5 shows the percentage of selected features (in terms of the mean and standard deviation) for all of the selected datasets. From this Table, it can be appreciated that the whole set of variables is used in most cases for datasets that have few variables, which indicates that there are no trivial variables for the classification and that MSCKTA is not performing an arbitrary selection. However, as the number of attributes grows, the number of selected attributes tends to decrease (note that in 6 of the datasets, the number of selected features is lower than a 50%).

Table 5 Percentage of features used for each dataset with MSCKTA and number of attributes for each dataset (a).

Dataset	a	Perc. of features	Dataset	a	Perc. of features
HA	3	100.00 ± 0.00	HE	19	68.42 ± 47.76
LI	4	100.00 ± 0.00	BA	19	52.63 ± 51.30
MA	5	100.00 ± 0.00	HC	22	59.09 ± 50.32
MO	6	100.00 ± 0.00	LA	29	27.59 ± 45.49
AP	7	85.71 ± 37.80	SI	33	80.65 ± 40.16
PI	8	62.50 ± 51.75	KR	38	44.74 ± 50.39
GL	9	88.89 ± 33.33	CR	43	57.14 ± 50.09
SA	9	77.78 ± 44.10	SP	44	47.73 ± 50.53
BW	9	100.00 ± 0.00	CA	51	55.10 ± 50.25
HY	13	84.62 ± 37.55	SO	60	36.67 ± 48.60
BR	15	80.00 ± 41.40	CO	60	38.33 ± 49.03
HO	16	50.00 ± 51.64	CG	61	47.46 ± 50.36

Note that the rest of algorithm-dependent estimators do not naturally perform feature selection due to the capacity control of SVM methods. However, for unregularised methods this could be an important characteristic to consider.

4.4 Analysis of different initialisation methods

Because of the local optimality of the iRprop$^+$ algorithm, several random or even fixed initial points for $\boldsymbol{\alpha}$ can be considered. For simplicity, the same initial point has been used for this optimisation in some previous works [10] (the initial point considered for all members of $\boldsymbol{\alpha}$ is 10^0, because it corresponds to the standard deviation of all of the variables in the dataset[4]). For the rest of experiments in this paper we thus considered $\alpha_i = 10^0$ in order to fairly compare to other methodologies. However, the suitable choice of these

[4] Note that a data standardisation procedure is applied before optimisation

initial points is a determining factor for the robustness of the optimisation. In order to analyse the stability of the algorithm with respect to this choice, we compare the results obtained from different initialisations (one initialisation per training/test set was used):

- Fixed initialisation with $\alpha_i = 10^0, i = 1, \ldots, d$.
- Random initialisation with $\alpha_i = 10^{r_i}, i = 1, \ldots, d, r_i \in [-1, 1]$.
- Random initialisation with $\alpha_i = 10^{r_i}, i = 1, \ldots, d, r_i \in [-3, 3]$.
- Deterministic distance-based initialisation proposed in subsection 3.4.

Table 6 shows the results of these initialisation procedures for 10 datasets (using the same experimental procedure than before) where it can be seen that the proposed distance-based strategy presents the most competitive performance (although close to the one obtained for $\alpha_i = 10^0$). From these results, it can be stated that both a random initialisation between $[-1, 1]$ or just initialising all $\alpha_i = 10^0$ result in a stable and robust optimisation performance (as opposed to initialise the random numbers between $[-3, 3]$, i.e. the cross-validation grid used). These results also show the possibility of initialising the problem in a more intelligent way, to further improve the results in those cases where the best possible performance is required. Note that the remaining methods shown in previous subsections could be also benefited from this initialisation. work.

Table 6 Results obtained from the different initialisations considered.

Dataset	$\alpha_i = 10^0$	$r_i \in [-1, 1]$	$r_i \in [-3, 3]$	Distance-based
mammographic	*84.70 ± 2.90*	*84.70 ± 2.90*	83.25 ± 3.70	**84.82 ± 2.73**
pima	77.73 ± 3.05	*77.86 ± 3.30*	64.85 ± 0.73	**77.87 ± 3.29**
glassG2	80.33 ± 10.68	*80.96 ± 12.41*	53.38 ± 2.74	**82.72 ± 11.02**
saheart	**74.69 ± 7.19**	70.34 ± 8.09	65.37 ± 0.31	*72.07 ± 7.55*
breast-w	**96.71 ± 2.34**	*96.42 ± 2.15*	94.71 ± 2.94	*96.42 ± 2.15*
heartY	**84.44 ± 7.15**	*80.74 ± 11.01*	55.55 ± 0.00	**84.44 ± 7.45**
breast	*71.34 ± 3.50*	70.28 ± 2.35	68.53 ± 3.65	**72.39 ± 3.69**
sick	*97.99 ± 1.20*	97.75 ± 1.01	97.69 ± 0.74	**98.06 ± 1.22**
credit-a	85.51 ± 4.21	*85.80 ± 3.79*	67.83 ± 9.03	**86.09 ± 4.28**
colic	**85.03 ± 5.82**	83.99 ± 7.11	63.87 ± 2.06	*84.48 ± 6.35*

4.5 Discussion

Several advantages of MSCKTA can be identified: algorithm independence, data distribution independence, simple optimisation, inherent feature selection, sparser SVM models, easy extension to the multiclass and regression learning tasks, to different types of kernels and when only the similarities between the patterns are available.

This last subsection is intended to provide a deeper analysis of the situations in which a multi-scale approach is useful. To provide this analysis, some scenarios in the benchmark datasets that were used are shown in Fig. 7, 8, 9, 10 and 11. For each figure, two of the original input dimensions have been selected and are represented together with the class labelling. Furthermore,

the kernel width that is associated with each dimension is included in the corresponding axis. These figures have been altered through the use of a random jitter methodology to better visualise the number of patterns per point. It is important to note how MSCKTA assigns equal α values to features with similar class geometry and $\alpha \to \infty$ values to non-relevant features.

Fig. 7 represents the case of a dataset that has two dimensions significant for classification (i.e., that have not been excluded by setting the associated kernel width to infinity); however these two dimensions present a different kernel width for each. Fig. 8 and 9 represent the case of a dataset with two significant dimensions for classification, which also presents similar widths. Fig. 10 shows the case in which one of the variables includes significant information and the other does not (i.e., the associated kernel value tends to infinity). Finally, Fig. 11 represents the case in which neither of the variables contains useful information about the labelling structure. A discussion of each case is included in the different figure captions.

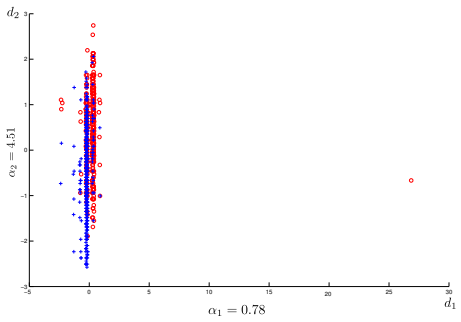


Fig. 7 Two-dimensional plot of the mammographic (MA) dataset (the first and second dimensions). In this case, the chosen kernel parameters for each data dimension vary significantly, which clarifies when a multi-scale kernel could be useful. Indeed, MSCKTA achieved better performance for this dataset (84.70%) than CKTA (82.41%).

Conclusions

This paper uses the centred kernel-target alignment concept to optimise a multi-scale kernel (considering a different width for each feature) using a gradient ascent algorithm. The optimisation of the kernel width has almost always been studied by means of learning machine execution, i.e., a simple "trial and error" procedure, which is computationally unaffordable for multiple kernel widths. The results obtained show that centred kernel-target alignment is highly correlated with performance and that the optimisation of a multi-scale kernel with this technique leads inherently to a better determined feature space, to feature selection, to significantly better results and to simpler models at a reasonable computational complexity. Moreover, a distance-based initilisation technique is presented which is able to further improve the results for

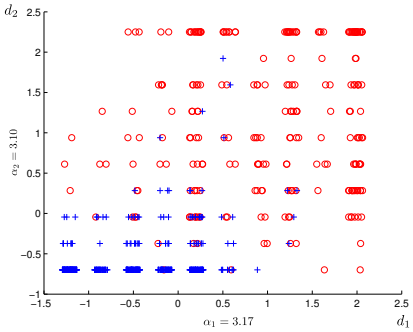


Fig. 8 Two-dimensional plot of the breast-w (BW) dataset (the first and second dimensions). Specifically, for this dataset almost the same kernel widths have been chosen for all of the dimensions. In this case, the performances of the CKTA and MSCKTA were similar (96.57% vs 96.71%, respectively). The graphical representation shows that the patterns can be differentiated by the use of a spherical kernel.

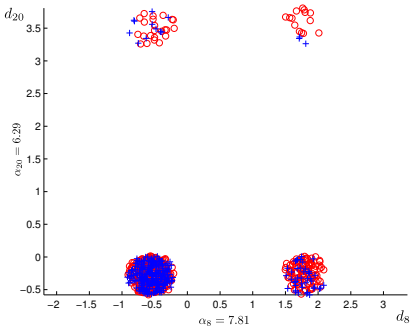


Fig. 9 Two-dimensional plot of the card (CA) dataset (8th and 20th dimensions). This figure represents the case of two dimensions used for the kernel computation, i.e., that contain useful information about the labelling structure of the data. Although these dimensions do not allow us to perfectly classify the data (note that the actual dimensionality of the dataset is 51), they give some useful discrimination knowledge about the patterns.

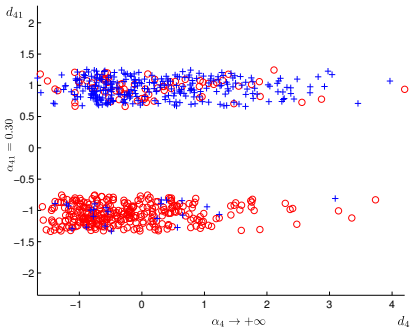


Fig. 10 Two-dimensional plot of the card (CA) dataset (4th and 41th dimensions). In this case, the plot represents one significant dimension and one that does not report useful information for classification.

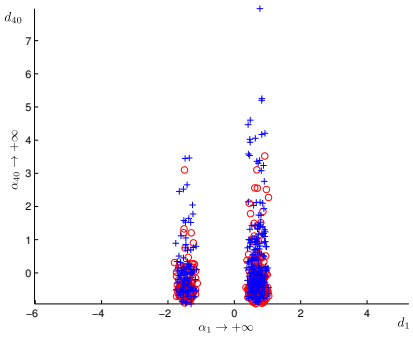


Fig. 11 Two-dimensional plot of the card (CA) dataset (first and 40th dimensions). This plot represents the case of two non-significant dimensions for the card dataset, where neither of the variables contains useful information about the labelling structure.

the majority of the datasets considered. Our results encourage the development of a hybrid metaheuristic approach with the gradient ascent method to explore the whole search space and obtain better results. Furthermore, as future work, a study of the multi-class and regression cases can be conducted to analyse whether the statements made in this paper are also valid for these learning paradigms.

Acknowledgments

This work has been partially subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P2011-TIC-7508 project of the "Junta de Andalucía" (Spain).

References

1. Ali, S., Smith-Miles, K.A.: A meta-learning approach to automatic kernel selection for support vector machines. Neurocomputing **70**(13), 173 – 186 (2006)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2007). URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`
3. Bartlett, P., Shawe–Taylor, J.: Generalization performance of support vector machines and other pattern classifiers. In: B. Schölkopf, C.J.C. Burges, A.J. Smola (eds.) Advances in Kernel Methods: Support Vector Learning, pp. 43–54. MIT Press (1999)
4. Bartlett, P.L., Mendelson, S.: Rademacher and gaussian complexities: risk bounds and structural results. Journal of Machine Learning Research **3**, 463–482 (2003)
5. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: D. Haussler (ed.) Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, pp. 144–152. ACM Press, Pittsburgh, PA (1992)
6. Bousquet, O., Elisseeff, A.: Stability and generalization. Journal of Machine Learning Research **2**, 499–526 (2002)
7. Bousquet, O., Herrmann, D.J.L.: On the complexity of learning the kernel matrix. In: In Advances in Neural Information Processing Systems 15, pp. 399–406. MIT Press (2003)
8. Braun, M.L., Buhmann, J.M., Müller, K.R.: On relevant dimensions in kernel feature spaces. J. Mach. Learn. Res. **9**, 1875–1908 (2008)

9. Chapelle, O., Rakotomamonjy, A.: Second order optimization of kernel parameters. In: Neural Information Processing Systems Workshop on Kernel Learning (NIPS) (2008)
10. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46**(1-3), 131–159 (2002)
11. Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for learning kernels based on centered alignment. Journal of Machine Learning Research **13**, 795–828 (2012)
12. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995)
13. Cristianini, N., Campbell, C., Shawe-taylor, J.: Dynamically adapting kernels in support vector machines. In: Advances in Neural Information Processing Systems 11, pp. 204–210. MIT Press (1998)
14. Cristianini, N., Kandola, J., Elisseeff, A., Shawe-Taylor, J.: On kernel-target alignment. In: Advances in Neural Information Processing Systems 14, pp. 367–373. MIT Press (2002)
15. Cuturi, M.: Fast global alignment kernels. In: Proceedings of the 28th International Conf. on Machine Learning (ICML-11), pp. 929–936 (2011)
16. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
17. Diosan, L., Rogozan, A., Pécuchet, J.P.: Improving classification performance of support vector machine by genetically optimising kernel shape and hyper-parameters. Appl. Intell. **36**(2), 280–294 (2012)
18. Do, H., Kalousis, A., Woznica, A., Hilario, M.: Margin and radius based multiple kernel learning. In: Proceedings of the European Conf. on Machine Learning and Knowledge Discovery in Databases: Part I, pp. 330–343. Springer-Verlag (2009)
19. Duan, K., Keerthi, Poo, A.N.: Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing **51**, 41–59 (2003)
20. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. In: Advances in Computational Mathematics, pp. 1–50. MIT Press (2000)
21. Fauvel, M.: Kernel matrix approximation for learning the kernel hyperparameters. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5418–5421 (2012)
22. Friedrichs, F., Igel, C.: Evolutionary tuning of multiple svm parameters. Neurocomputing **64**, 107–117 (2004). Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004
23. Gai, K., Chen, G., Zhang, C.: Learning kernels with radiuses of minimum enclosing balls. In: Proceedings of the International Conference on Neural Information Processing Systems (NIPS), pp. 649–657. Curran Associates, Inc. (2010)
24. Gascón-Moreno, J., Ortiz-García, E.G., Salcedo-Sanz, S., Paniagua-Tineo, A., Saavedra-Moreno, B., Portilla-Figueras, J.A.: Multi-parametric gaussian kernel function optimization for ε-svmr using a genetic algorithm. In: Proc. of the 11th Intern. Conf. on Artificial neural networks, *IWANN'11*, vol. 2, pp. 113–120. Springer-Verlag (2011)
25. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation **7**, 219–269 (1995)
26. Glasmachers, T.: Gradient based optimization of support vector machines. Ph.D. thesis (2008)
27. Gönen, M., Alpaydin, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
28. Guermeur, Y., Lifchitz, A., Vert, R.: Kernel for protein secondary structure prediction, p. 416. The MIT Press, Cambridge, Massachussetts (2004)
29. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2), 159–195 (2001)
30. Howard, A., Jebara, T.: Transformation learning via kernel alignment. In: Proceedings of the 2009 International Conference on Machine Learning and Applications, pp. 301–308. IEEE Computer Society, Washington, DC, USA (2009)
31. Howley, T., Madden, M.G.: The genetic kernel support vector machine: Description and evaluation. Artificial Intelligence Review **24**, 379–395 (2005)
32. Igel, C., Glasmachers, T., Mersch, B., Pfeifer, N., Meinicke, P.: Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. IEEE/ACM Trans. Comput. Biol. Bioinformatics **4**(2), 216–226 (2007)

33. Igel, C., Hüsken, M.: Empirical evaluation of the improved rprop learning algorithms. Neurocomputing **50**, 105–123 (2003)
34. Jaakkola, T.S., Haussler, D.: Probabilistic Kernel Regression Models. In: Proceedings of the 1999 Conference on AI and Statistics (1999)
35. Kandola, J., Shawe-Taylor, J., Cristianini, N.: On the extensions of kernel alignment. Technical report, http://www.neurocolt.org (2002)
36. Keerthi, S., Sindhwani, V., Chapelle, O.: An efficient method for gradient-based adaptation of hyperparameters in svm models. In: B. Schölkopf, J. Platt, T. Hoffman (eds.) Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA (2007)
37. Kim, S.J., Magnani, A., Boyd, S.: Optimal kernel selection in kernel fisher discriminant analysis. In: Proceedings of the 23rd international conference on Machine learning, ICML '06, pp. 465–472. ACM, New York, NY, USA (2006)
38. Koltchinskii, V.: Rademacher penalties and structural risk minimization. IEEE Transactions on Information Theory **47**(5), 1902–1914 (2001)
39. Kwok, J.T., Tsang, I.W.: Learning with idealized kernels. In: Proceedings of the Twentieth International Conference (ICML), pp. 400–407. AAAI Press (2003)
40. Lanckriet, G.R.G., Cristianini, N., Bartlett, P.L., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research **5**, 27–72 (2004)
41. Opper, M., Winther, O.: Gaussian processes for classification: Mean-field algorithms. Neural Comput. **12**(11), 2655–2684 (2000)
42. Ortiz-García, E.G., Salcedo-Sanz, S., Pérez-Bellido, A.M., Portilla-Figueras, J.A.: Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions. Neurocomput. **72**(16-18), 3683–3691 (2009)
43. Pérez-Ortiz, M., Gutiérrez, P.A., Sánchez-Monedero, J., Hervás-Martínez, C.: Multi-scale Support Vector Machine Optimization by Kernel Target-Alignment. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), pp. 391–396 (2013)
44. Phienthrakul, T., Kijsirikul, B.: Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In: Proceedings of the 2005 conf. on Genetic and evolutionary computation, GECCO '05, pp. 905–911 (2005)
45. Pothin, J.B., Richard, C.: A greedy algorithm for optimizing the kernel alignment and the performance of kernel machines. In: Proceedings on the European Signal Processing Conference (2006)
46. Ramona, M., Richard, G., David, B.: Multiclass feature selection with kernel gram-matrix-based criteria. IEEE Trans. Neural Netw. Learning Syst. **23**(10), 1611–1623 (2012)
47. Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Transactions on Neural Networks **10**, 1000–1017 (1999)
48. Shamsheyeva, A., Sowmya, A.: The anisotropic gaussian kernel for svm classification of hrct images of the lung. In: Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004, pp. 439 – 444 (2004)
49. Smola, A.J., Schölkopf, B., Müller, K.R.: The connection between regularization operators and support vector kernels. Neural Networks **11**(4), 637–649 (1998)
50. Soares, C., Brazdil, P.B.: Selecting parameters of svm using meta-learning and kernel matrix-based meta-features. In: Proceedings of the ACM symposium on Applied computing, pp. 564–568 (2006)
51. Sollich, P.: Probabilistic methods for support vector machines. In: Advances in Neural Information Processing Systems 12, pp. 349–355. MIT Press (2000)
52. Sollich, P.: Bayesian methods for support vector machines: Evidence and predictive class probabilities. Machine Learning **46**, 21–52 (2002)
53. Srebro, N., Ben-david, S.: Learning bounds for support vector machines with learned kernels. In: In Annual Conference On Learning Theory (COLT, pp. 169–183. Springer (2006)
54. Tipping, M.E.: The relevance vector machine. In: Proceedings of the International Conference on Neural Information Processing Systems (NIPS), pp. 652–658 (1999)

55. Vapnik, V., Chapelle, O.: Bounds on error expectation for support vector machines. Neural Comput. **12**(9), 2013–2036 (2000)
56. Vapnik, V.N.: Statistical learning theory, 1 edn. Wiley (1998)
57. Vapnik, V.N., Chervonenkis, A.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications **16**(2), 264–280 (1971)
58. Wahba, G., Wahba, G., Lin, Y., Lin, Y., Zhang, H., Zhang, H.: Generalized approximate cross validation for support vector machines, or, another way to look at margin-like quantities. In: Advances in Large Margin Classi, pp. 297–309. MIT Press (1999)
59. Wolpert, D., Macready, W.: No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on **1**(1), 67–82 (1997)
60. Wu, K.P., Wang, S.D.: Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. Pattern Recogn. **42**(5), 710–717 (2009)
61. Xiong, H., Swamy, M.N.S., Ahmad, M.O.: Optimizing the kernel in the empirical feature space. IEEE Transactions on Neural Networks **16**(2), 460–474 (2005)
62. You, D., Hamsici, O.C., Martinez, A.M.: Kernel optimization in discriminant analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(3), 631–638 (2011). DOI http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.173

4.2. Kernelising the proportional odds model through kernel learning techniques

As stated before, one of the most widely used ordinal regression algorithms in real-world problems is the proportional odds model (POM), despite the linearity of the resultant decision boundaries, which hampers the obtainment of a well-suited classifier. Through different proposals, the following paper explores the notions of kernel trick and empirical feature space to reformulate the POM method and obtain nonlinear decision boundaries. Additionally, we also propose a new technique for aligning the kernel matrix taking into account the ordinal information (i.e. an ordinal kernel learning methodology) as well as a regularised gradient-ascent methodology which is used to select the optimal dimensionality for the empirical feature space. Note that optimally selecting the dimensionality of the empirical feature space is important, firstly for visualisation purposes and secondly as a regularisation technique. In contrast to support vector machines, logistic regression does not directly perform a regularisation step, so the use of this method is necessary (also to reduce the number of dimensions and alleviate the computational load). All the methods proposed in this paper are independent of the learning algorithm and could be employed in conjunction with any classifier.

The different experiments show that the proposed kernel techniques are able to increase the performance of linear ordinal regression methods, such as the POM, and reach the performance of the state-of-the-art methods, while still being able to derive natural probability estimates. Moreover, the gradient-ascent method could be used to approximately visualise the effect of the kernel function on the data and the optimal projection to a subspace which maintains the ordinal information.

Kernelising the Proportional Odds Model through Kernel Learning techniques

M. Pérez-Ortiz[a], P.A. Gutiérrez[a], M. Cruz-Ramírez[a], J. Sánchez-Monedero[a], C. Hervás-Martínez[a]

[a] *University of Córdoba, Dept. of Computer Science and Numerical Analysis*
Rabanales Campus, Albert Einstein building, 14071 - Córdoba, Spain

Abstract

The classification of patterns into naturally ordered labels is referred to as ordinal regression, which is a very common setting for real world applications. One of the most widely used ordinal regression algorithms is the Proportional Odds Model (POM), despite the linearity of the resultant decision boundaries. Through different proposals, this paper explores the notions of kernel trick and empirical feature space to reformulate the POM method and obtain nonlinear decision boundaries. Moreover, a new technique for aligning the kernel matrix taking into account the ordinal problem information is proposed, as well as a regularised gradient ascent methodology which is used to select the optimal dimensionality for the empirical feature space. The capability of the different developed methodologies is evaluated by the use of a nonlinearly separable toy dataset and an extensive set of experiments over 28 ordinal datasets. The results indicate that the tested methodologies are competitive with respect to other state-of-the-art algorithms, and they significantly improve the original POM algorithm.

Keywords:
Proportional Odds Model, Ordered logit, Ordinal Regression, Ordinal Classification, Kernel Trick, Kernel Learning

1. Introduction

In this paper, we consider the specific problem of ordinal regression, which shares properties of classification and regression settings. Formally, $\mathcal{Y}$ (the target space) is a finite set, but there exists an ordering among its elements. In contrast to regression, $\mathcal{Y}$ is a non-metric space, thus distances among categories are unknown. Besides, the zero-one loss function usually considered for standard classification does not reflect the ordering of $\mathcal{Y}$. Ordinal regression (or classification) problems arise in fields as information retrieval, preference learning, economy, and statistics, forming an emerging field in the areas of machine learning and pattern recognition.

A great number of statistical methods for categorical data treat all response variables as nominal, in such a way that the results are invariant to category permutations on those variables. However, there are many advantages in treating an ordered categorical variable as ordinal rather than nominal [1, 2]. In this vein, several approaches to tackle ordinal regression have been proposed in the domain of machine learning over the years, the Proportional Odds Model (POM) being one of the first ones, dating back to 1980 [3]. Indeed, the POM can be contextualised in the most popular framework for ordinal regression, *i.e.*, the threshold models [4, 5, 3], which are based on the assumption that an underlying real-valued outcome exists (also known as latent variable), although it is unobservable. These methods try to determine the nature of the underlying outcome by using a function $f(\cdot)$ and a set of thresholds to represent intervals in the range of $f(\cdot)$. Although very sophisticated and successful learning techniques have been recently developed for ordinal regression [6, 4, 5, 7], the use of the POM method is widespread. However, the resulting decision boundaries are linear, which is an unrealistic assumption for many real world problems. To deal with this issue, the proposals presented in this paper make use of the notion of the so-called kernel trick, which implicitly maps input patterns into a high-dimensional feature space via a function $\Phi(\cdot)$ in order to compute nonlinear decision boundaries. The standard process for applying the kernel trick requires reformulating the learning algorithm based on dot products between the different training points, which implies some difficulties in the case of the POM, as we will see. Alternatively, we consider the Empirical Feature Space (EFS) [8, 9], which preserves the geometrical structure of the original feature space (the dot products of the corresponding images are equal to the original kernel values, and the distances and angles in the feature space are uniquely determined by dot products). The EFS is Euclidean, this allowing the kernelisation of all kinds of linear machines [10, 11], with the advantage that the algorithm does not need to be formulated to deal with dot products.

The dimensionality of the EFS is the rank of the kernel matrix, which can be very high (*e.g.*, in the case of a Gaussian kernel it usually corresponds to the number of training patterns). This is a key factor in the reformulation of the POM algorithm, whose computational cost is closely related to the dimensionality of the dataset. Therefore, we propose different techniques to control this dimensionality while approximating the original

*This paper has been invited to be included in the "Special Issue Neurocomputing-IWANN2013".

information contained in the kernel matrix and therefore including some form of regularisation.

On the other hand, the performance of the POM model constructed in the EFS directly depends on how well the kernel function is adapted to the problem considered. Because of this, kernel-target alignment, a well-known kernel learning technique, [12, 13] is considered in this paper to better adapt the EFS to each dataset. This technique is extended by including ordinal weights in order to take the ordinal nature of the target variable into account. As will be analysed, such a kernel learning technique is very useful for creating a method to automatically compute the dimensionality of the EFS.

Summarising, the contributions of the paper can be said to be threefold: 1) the application of the EFS to compute nonlinear decision boundaries for the POM at a limited computational cost, leading naturally to probabilistic outputs; 2) an ordinal kernel learning technique to better match the different datasets; 3) an extension of this kernel learning technique in order to automatically decide the dimensionality of the EFS.

The kernelisation of the POM has also been considered in [14], where the POM method is extended for non-crisp ordinal regression task. Maximisation of the regularised loss function is accomplished by considering the representer theorem [15]. However, the paper makes reference to a different setting, where partial class memberships are given for the patterns, while we are provided with crisp ordinal targets. Moreover, our method approaches the optimisation of the model in a more direct way by redefining the model in the EFS. Other works have considered before a nonlinear version of the POM method (or more generally, a nonlinear version of logistic regression) by the use of artificial neural networks [16] or by including polynomial combinations of the input features. However, these strategies imply difficult optimisation processes. As will be shown in the experimental section, the use of the EFS with a Gaussian kernel allows the POM method to obtain much better results and to handle nonlinear decision boundaries. The experiments also show that the selection of the optimal dimensions is a crucial step which can significantly improve the algorithm performance, as well as the inclusion of the ordinal information in the kernel optimisation process.

The rest of the paper is organised as follows: Section 2 presents some useful previous notions; Section 3 shows a description of the different proposals; Section 4 describes the experimental study and analyses the results; and finally, Section 5 outlines some conclusions and future work.

2. Previous notions

The goal in classification is to assign an input vector $\mathbf{x}$ to one of Q discrete classes $C_q, q \in \{1, \dots, Q\}$. A formal framework for the ordinal regression problem can be introduced considering an input space $\mathcal{X} \in \mathbb{R}^{m \times d}$, where m is the number of training patterns and d is the data dimensionality. Moreover, an outcome space $\mathcal{Y} = \{C_1, C_2, \dots, C_Q\}$ can be defined, where the labels are ordered in such a way that $C_1 \prec C_2 \prec \cdots \prec C_Q$, where $\prec$ denotes the order relation. The objective for this learning setting is to find a prediction rule $f : \mathcal{X} \to \mathcal{Y}$ by using an i.i.d. training sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^m \in \mathcal{X} \times \mathcal{Y}$. The following subsections describe some of the concepts needed to understand the methodology proposed in this paper.

2.1. Proportional Odds Model

This is one of the first models specifically designed for ordinal regression, and it arises from a statistical background [3]. Let h denote an arbitrary monotonic link function and $P(y \preceq C_q|\mathbf{x})$ the probability that a pattern $\mathbf{x}$ belongs to a class lower to C_q (in the ordinal scale). The model:

$$h\Big(P(y \preceq C_q|\mathbf{x})\Big) = b_q - \boldsymbol{\beta}^\top \mathbf{x}, \quad q = 1, \dots, Q-1, \qquad (1)$$

links the cumulative probabilities to a linear predictor and imposes an stochastic ordering of the space $\mathcal{X}$, where b_q is the threshold separating classes C_q and C_{q+1} and $\boldsymbol{\beta}$ is a linear projection. This model is naturally derived from the latent variable motivation; then instead of fitting a decision rule $f : \mathcal{X} \to \mathcal{Y}$ directly, this model defines a probability density function over the class labels for a given feature vector $\mathbf{x}$. Let us assume that the ordinal response comes from a coarsely measured latent continuous variable $f(\mathbf{x})$. Thus, label C_q in the training set is observed if and only if $f(\mathbf{x}) \in [b_{q-1}, b_q]$, where the function f (latent utility) and $b = \{b_0, b_1, ..., b_{Q-1}, b_Q\}$ are determined from data. By definition, $b_0 = -\infty$ and $b_Q = +\infty$ and the real line $f(\mathbf{x})$ is divided into Q consecutive intervals, where each interval corresponds to a category C_q.

Now, let define a model of the latent variable, $f(\mathbf{x}) = \boldsymbol{\beta}^\top \mathbf{x} + \epsilon$, where ϵ is the random variable with zero expectation, $\mathbf{E}[\epsilon] = 0$, and distributed according to the distribution function F_ϵ. Then, it follows that:

$$P(y \preceq C_q|\mathbf{x}) = \sum_{k=1}^q P(y = C_k|\mathbf{x}) = \sum_{k=1}^q P(f(\mathbf{x}) \in [b_{k-1}, b_k]) =$$

$$= P(f(\mathbf{x}) \in [-\infty, b_q]) = P(\boldsymbol{\beta}^\top \mathbf{x} + \epsilon \leq b_q) = P(\epsilon \leq b_q - \boldsymbol{\beta}^\top \mathbf{x}) =$$

$$= F_\epsilon(b_q - \boldsymbol{\beta}^\top \mathbf{x}).$$

If a distribution assumption F_ϵ is made for ϵ, the cumulative model is obtained by choosing, as the inverse link function h^{-1}, the inverse distribution F_ϵ^{-1} (quantile function). Note that $F_\epsilon^{-1} : [0, 1] \to (-\infty, +\infty)$ is a monotonic function. The most common choice for F_ϵ is the logistic function [3].

2.2. Ideal kernel

Let $\mathcal{H}$ denote a high-dimensional or infinite-dimensional Hilbert space. Then, for any mapping of patterns $\Phi : \mathcal{X} \to \mathcal{H}$, the inner product $k(\mathbf{x}_i, \mathbf{x}_j) = \Big\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \Big\rangle_{\mathcal{H}}$ of the mapped inputs is known as a kernel function, giving rise to a positive semidefinite (PSD) matrix $\mathbf{K}$ for a given input set $\mathcal{X}$.

Although properties of a kernel function k are important, often the kernel matrix ($\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$) plays a more important role than the kernel function, given that most kernel algorithms work with this matrix. Kernel matrices contain information about the similarity among the patterns in a dataset. Therefore, the empirical ideal kernel [13], $\mathbf{K}^*$, (*i.e.*, the matrix that

would represent perfect similarity information) will submit the following structure:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 & \text{if } y_i = y_j, \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

where $\mathbf{K}^*_{ij} = k^*(\mathbf{x}_i, \mathbf{x}_j)$. Roughly speaking, $\mathbf{K}^*$ provides information about which patterns in the dataset should be considered as similar when performing some learning task. As we are dealing with a classification problem, patterns from the same class should be considered totally similar, while patterns from other classes should be considered as different as possible.

2.3. Centered kernel-target alignment

Suppose an ideal kernel matrix $\mathbf{K}^*$ and a given real kernel matrix $\mathbf{K}$. The underlying idea for kernel-target alignment (KTA) [13] is to choose the kernel matrix $\mathbf{K}$ (among a set of different matrices) closest to the ideal matrix $\mathbf{K}^*$. This can be evaluated by the Frobenius inner product between these matrices (*i.e.*, $\langle \mathbf{K}, \mathbf{K}^* \rangle_{\mathrm{F}} = \sum_{i,j=1}^{m} k(\mathbf{x}_i, \mathbf{x}_j) \cdot k^*(\mathbf{x}_i, \mathbf{x}_j)$), which give us information of how well the patterns are classified in his own category. Indeed, if we consider Eq. (2), the Frobenius inner product could be rewritten as $\langle \mathbf{K}, \mathbf{K}^* \rangle_{\mathrm{F}} = \sum_{y_i = y_j} k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j)$, where the term $\sum_{y_i = y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ is related to the within-class distance, and the term $\sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ is related to the between-class distance.

The KTA between two kernel matrices $\mathbf{K}$ and $\mathbf{K}^*$ is defined as:

$$\mathcal{A}(\mathbf{K}, \mathbf{K}^*) = \frac{\langle \mathbf{K}, \mathbf{K}^* \rangle_{\mathrm{F}}}{\sqrt{\langle \mathbf{K}^*, \mathbf{K}^* \rangle_{\mathrm{F}} \langle \mathbf{K}, \mathbf{K} \rangle_{\mathrm{F}}}}. \quad (3)$$

This quantity is totally maximised when the kernel function is capable to reflect the properties of the training dataset used to define the ideal kernel matrix.

However, some problems are found when considering KTA for datasets with skewed class distributions [13, 17]. These problems can be solved by the use of centred kernel matrices [12], leading a methodology (centred kernel-target alignment, CKTA) that have demonstrated to correlate better with performance than with the original definition of KTA. CKTA basically extends KTA by centring the patterns in the feature space. The centred kernel version of a matrix $\mathbf{K}$ can be written as:

$$\mathbf{K}_c = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{m}} - \mathbf{1}_{\frac{1}{m}}\mathbf{K} + \mathbf{1}_{\frac{1}{m}}\mathbf{K}\mathbf{1}_{\frac{1}{m}},$$

where $\mathbf{1}_{\frac{1}{m}}$ corresponds to a matrix with all the elements equal to $\frac{1}{m}$. $\mathbf{K}_c$ will also be a PSD matrix, fulfilling $k(\mathbf{x}, \mathbf{x}) \geq 0 \;\; \forall\, \mathbf{x} \in \mathcal{X}$ and symmetry.

2.4. Empirical Kernel Mapping

In this section, the Empirical Feature Space (EFS) [8] spanned by the training data is defined. By definition, a kernel matrix $\mathbf{K}$ can be diagonalised as follows:

$$\mathbf{K}_{(m \times m)} = \mathbf{P}_{(m \times r)} \cdot \mathbf{\Lambda}_{(r \times r)} \cdot \mathbf{P}^{\top}_{(r \times m)}, \quad (4)$$

where r is the rank of $\mathbf{K}$, $\mathbf{\Lambda}$ is a diagonal matrix containing the r non-zero eigenvalues of $\mathbf{K}$ in decreasing order (*i.e.*, $\lambda_1, \ldots, \lambda_r$), and $\mathbf{P}$ is a matrix consisting of the eigenvectors associated to those r eigenvalues (*i.e.*, $\mathbf{u}_1, \ldots, \mathbf{u}_r$) in such a way that $\mathbf{K} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{u}_i^T$. Note that this mapping corresponds to the principal component analysis *whitening* step [18], but applied to the kernel matrix, instead of the covariance one. Then, the EFS can be defined as an Euclidean space preserving the dot product information about $\mathcal{H}$ contained in $\mathbf{K}$ (*i.e.*, this space is isomorphic to the embedded feature space $\mathcal{H}$, but being Euclidean). Since distances and angles of the vectors in the feature space are uniquely determined by dot products, the training data have the same geometrical structure in both the EFS and the feature space. The map from the input space to this r-dimensional EFS is defined as $\Phi_r^e : \mathcal{X} \to \mathbb{R}^r$. More specifically:

$$\Phi_r^e : \mathbf{x}_i \to \mathbf{\Lambda}^{-1/2} \cdot \mathbf{P}^{\top} \cdot (k(\mathbf{x}_i, \mathbf{x}_1), \ldots, k(\mathbf{x}_i, \mathbf{x}_m))^{\top}. \quad (5)$$

It can be checked that the kernel matrix of training images obtained by this transformation corresponds to $\mathbf{K}$, when considering the standard dot product [8, 9].

Furthermore, the EFS provides us with the opportunity to limit the dimensionality of the space by choosing the $j \leq r$ dominant eigenvalues (and their associated eigenvectors) to project the data, while maintaining the most important part of the structure of $\mathcal{H}$. Nevertheless, how to correctly choose j is still a difficult issue to be solved.

Figure 1 has been included in order to graphically clarify the concept of EFS. It can be seen that, despite the fact that the three most representative dimensions are not enough to linearly separate the data, they actually provide useful information about the order of the classes and the separation between them.

3. Proposed methodology: Tackling the ordinal information via the kernel trick

Although Eq. (1) could be directly kernelised in the same vein that it is done with support vector machines (SVMs) (see, for example, [19]), this would imply substituting the standard hinge loss by the negative log likelihood loss (in our case, both adapted to ordinal regression). Because of the nature of this log likelihood loss function, this would reduce the sparsity of the obtained kernel machine. The reason then to consider the EFS is precisely to be able to reduce the dimensionality of the obtained kernel machine by the method presented in Section 3.2.1, which, in general, should improve the generalisation performance.

Three differentiated proposals can be found in this section of the paper. Firstly, we propose how to extend the POM method to deal with a nonlinear transformation of the input variables making use of the kernel trick (*i.e.*, the above mentioned EFS). Secondly, we reformulate the notion of CKTA (a common strategy for kernel learning) to deal with classification problems that present an ordinal structure by imposing different weights for the different similarity errors. Finally, a new method is proposed for reducing the dimensionality of the subspace to which the data are projected. As said before, this is very useful for the reformulated POM, because it can decrease a lot the computational complexity (as opposed to considering the EFS with the full-rank decomposition).

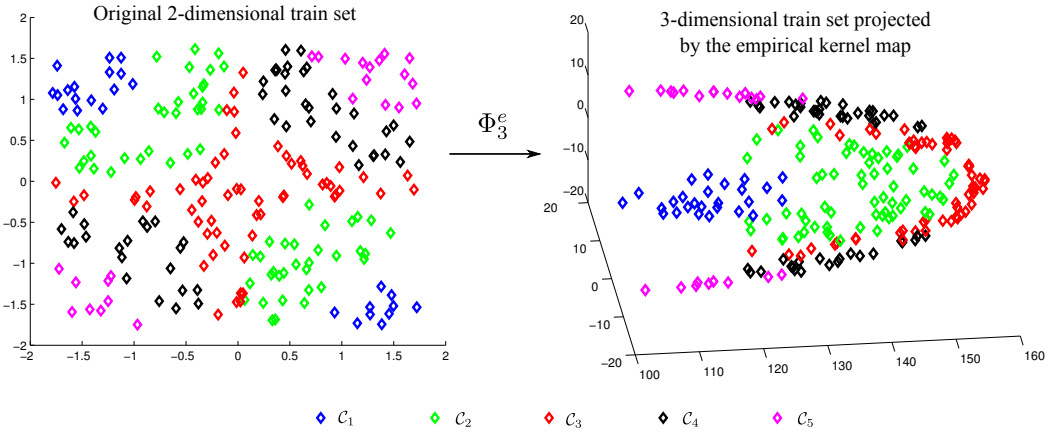


Figure 1: 3-dimensional approximation of the EFS induced by a Gaussian kernel for the nonlinearly separable synthetic toy dataset.

3.1. Proportional Odds Model in the Empirical Feature Space

Far beyond the definition of the EFS, it is well-known that the kernel trick turns a linear decision boundary in $\mathcal{H}$ into a nonlinear decision boundary in $\mathcal{X}$. This allows the formulation of nonlinear variants of many algorithms (those which can be cast in terms of the inner products between patterns). When using the EFS, this last restriction is avoided and any standard linear decision algorithm can be used, without any loss of generality. Figure 2 shows the case of a synthetic dataset representing a nonlinearly separable classification task and its transformation to the two-dimensional EFS (using the two most dominant eigenvectors), which is linearly separable.

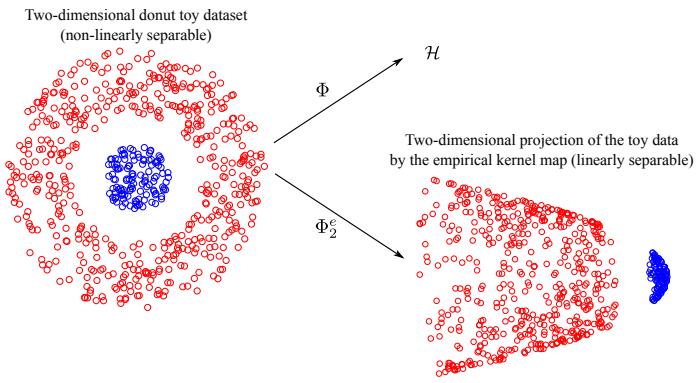


Figure 2: Synthetic two-dimensional dataset representing a nonlinearly separable classification problem and its transformation to the 2 dominant dimensions of the EFS induced by the Gaussian kernel function (linearly separable problem). Note that the $\mathcal{H}$ space can not be represented itself. However, the transformation performed when applying the kernel trick can be observed by analysing the two-dimensional EFS representation.

Now, consider the use of the EFS transformation $\Phi_r^e(\mathbf{x})$ (Eq. (5)) for redefining the POM. Eq. (1) is reformulated as:

$$h\left(P(y \preceq C_q|\mathbf{x})\right) = b_q - \boldsymbol{\beta}^\top \Phi_r^e(\mathbf{x}) = \tag{6}$$

$$= b_q - \boldsymbol{\beta}^\top \boldsymbol{\Lambda}^{-1/2} \cdot \mathbf{P}^\top \cdot (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_m))^\top. \tag{7}$$

In this case, the model of the latent variable will submit the formulation $f(\Phi_r^e(\mathbf{x})) = \boldsymbol{\beta}^\top \cdot \Phi_r^e(\mathbf{x}) + \epsilon$, where $\boldsymbol{\beta}$ will be a linear projection. However, this projection will perform as a nonlinear decision function in $\mathcal{X}$, since a nonlinear transformation of the input variables is being used.

3.2. Kernel-Target Alignment for ordinal classification

Standard multinomial classification problems have been studied by using KTA based on a geometrical interpretation [20], resulting in a simple modification of the original KTA (which was initially designed for binary problems). Instead of considering the kernel equal to -1 when the patterns do not belong to the same class, it is assigned to $-1/(Q-1)$, being Q the number of classes in the problem. This is done because each description $\mathbf{x}$ is associated to one of the Q vertices of a $(Q-1)$-dimensional centred simplex. However, such approach is not consistent when considering a dataset with an ordinal structure, because all the errors committed are equally weighted and all the classes are said to be equally similar to the rest of classes.

For the sake of understanding, consider a dataset D composed of five patterns belonging to four different classes, *i.e.*, $D = \{(\mathbf{x}_1, C_1), (\mathbf{x}_2, C_2), (\mathbf{x}_3, C_3), (\mathbf{x}_4, C_3), (\mathbf{x}_5, C_4)\}$. The ideal kernel matrix for D can be seen in Table 1. Bold face is used in this Table to outline some of the entries of these matrices. Note that the kernel matrix can be seen from a pattern similarity/dissimilarity perspective. Now, examine the two arbitrary kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$. In the Gram matrix $\mathbf{K}_1$, the pattern $\mathbf{x}_1 \in C_1$ is said to be similar to $\mathbf{x}_2 \in C_2$, while, in the Gram matrix $\mathbf{K}_2$, it is said to be similar to $\mathbf{x}_5 \in C_4$. In the case of ordinal regression, those misclassification errors involving a higher number of categories between the real label and the predicted one (in the ordinal scale) should be more penalised [2, 21, 22]. Similarly, matrix $\mathbf{K}_2$ (which is confusing a pattern from the first class with one of the fourth one) should result in a lower KTA than $\mathbf{K}_1$ (which is confusing this pattern with one of the neighbouring classes).

Table 2 shows three different error weighting cost matrices used in previous works. The first one is associated with the nominal classification setting, where all the misclassification errors are considered to be equal. The second one, is known as the absolute cost matrix, and it takes into account the difference of the assigned values for the categories, that is $|r(y_j) - r(y_i)|$, $(r(y_i)$ being the ranking for a given target y_i, *i.e.*, the position of y_i in

Table 1: Example of different kernel matrices for the hypothetical dataset D.

Ideal kernel matrix $\mathbf{K}^*$					Kernel matrix $\mathbf{K}_1$					Kernel matrix $\mathbf{K}_2$				
+1	−1	−1	−1	−1	+1	**+1**	−1	−1	−1	+1	−1	−1	−1	**+1**
−1	+1	−1	−1	−1	**+1**	+1	−1	−1	−1	−1	+1	−1	−1	−1
−1	−1	+1	+1	−1	−1	−1	+1	+1	−1	−1	−1	+1	+1	−1
−1	−1	+1	+1	−1	−1	−1	+1	+1	−1	−1	−1	+1	+1	−1
−1	−1	−1	−1	+1	−1	−1	−1	−1	+1	**+1**	−1	−1	−1	+1

the ordinal scale). Finally, the third one is the quadratic version of the absolute cost matrix. Absolute cost and quadratic absolute cost are commonly considered for ordinal regression problems, as a way of obtaining classifiers which minimise those misclassification errors involving several categories in the ordinal scale.

Table 2: Different cost matrices which can be found in the literature.

	Nominal cost				Absolute cost				Quadratic abs. cost			
	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
C_1	0	1	1	1	0	1	2	3	0	1	4	9
C_2	1	0	1	1	1	0	1	2	1	0	1	4
C_3	1	1	0	1	2	1	0	1	4	1	0	1
C_4	1	1	1	0	3	2	1	0	9	4	1	0

In the same vein, we propose to consider these matrices when obtaining the KTA of a matrix, in order to penalise differently the misalignment errors of an evaluated matrix. That is, a weighting matrix $\mathbf{W}$ is defined in such a way that $\mathbf{K}^* \circ \mathbf{W}$ imposes a weighting for the different similarity or dissimilarity errors committed, where $\mathbf{A} \circ \mathbf{B}$ represents the hadamard or entrywise product between matrices $\mathbf{A}$ and $\mathbf{B}$. A first idea for weighting errors would be the use of the absolute errors commonly used for ordinal classification, *i.e.*:

$$w(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } y_i = y_j, \\ |r(y_i) - r(y_j)|, & \text{otherwise.} \end{cases} \tag{8}$$

As discussed previously, the centred version of the matrices will be considered, avoiding problems with skewed class distributions. Therefore, the proposed ordinal version $\tilde{\mathcal{A}}_c$ of CKTA is defined as follows:

$$\tilde{\mathcal{A}}_c(\mathbf{K}, \mathbf{K}^*) = \mathcal{A}_c(\mathbf{K}, \mathbf{K}^* \circ \mathbf{W}). \tag{9}$$

This reformulation of CKTA for ordinal problems can be used for optimising the parameters of the kernel matrix (subsection 3.2.1), as well as for choosing the optimal dimensions for projecting the data onto a lower dimensional space (subsection 3.3). Both approaches will be considered for the experiments in order to improve the quality of the EFS in conjunction with the POM method.

3.2.1. *Optimisation of the ordinal Centred Kernel-Target Alignment via Multiple Kernel Learning*

For the optimisation of the proposed ordinal CKTA, one could use any of the optimisation strategies proposed for the original CKTA. In this paper, we will use two different strategies. This subsection presents one of them, and subsection 3.3 proposes the other. In this subsection, we use a Quadratic Programming problem (QP) (by means of multiple kernel learning techniques), which has a single global maximum and it is easier to optimise. The solution of this QP problem will result in a kernel matrix defining the optimal EFS for the considered problem. We optimise a convex combination of kernel matrices, where each matrix is associated to a different parameter for the kernel width. Therefore, we fix a set of p possible parameter values for the kernel width α, *i.e.*, $\{\alpha_1, \ldots, \alpha_p\}$ and compute the kernel matrices obtained for these values $\{\mathbf{K}_{\alpha_1}, \ldots, \mathbf{K}_{\alpha_p}\}$. To optimise CKTA (or the proposed ordinal version), we can derive a kernel matrix $\mathbf{K}_{\boldsymbol{\delta}} = \sum_{i=1}^{p} \delta_i \mathbf{K}_{\alpha_i}$ with $\delta_i \geq 0$ and $\sum_{i=1}^{p} \delta_i = 1$. The optimisation problem for the ordinal version of CKTA will be the following:

$$\max_{\boldsymbol{\delta} \in \mathcal{M}} \frac{\langle \mathbf{K}_{\boldsymbol{\delta}_c}, \mathbf{K}^* \circ \mathbf{W} \rangle_{\mathrm{F}}}{\|\mathbf{K}_{\boldsymbol{\delta}_c}\|_{\mathrm{F}}},$$

where $\|\mathbf{A}\|_{\mathrm{F}} = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_{\mathrm{F}}}$ and $\mathcal{M} = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_2 = 1\}$. The QP optimization problem associated can be solved as in [12].

To show how the different weights in Table 1 may influence the choice of the parameters we include the optimisation surfaces obtained for $\boldsymbol{\delta}$ when $\boldsymbol{\alpha} = \{0.01, 1, 100\}$. We use a three-dimensional simplex (to fulfil $\delta_i \geq 0$ and $\sum_{i=1}^{3} \delta_i = 1$), as can be seen in Figure 3, where the coloured points show how to select the values of the parameters $\boldsymbol{\delta}$ in order to fulfil their constraints. Figure 4 shows these optimisation surfaces for two datasets of the experiments considered in this paper (LEV and toy) and the three weight matrices in Table 1. As can be appreciated, the surfaces are very different and the optimum value can be found in a different region of the simplex.

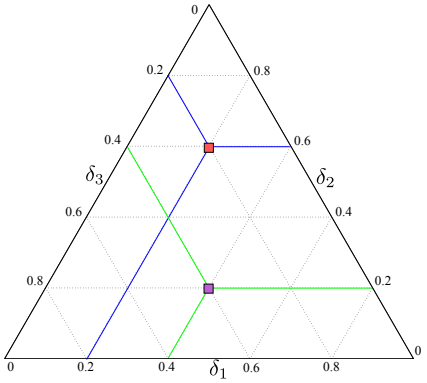


Figure 3: Simplex example where it can be seen how to compute δ_1, δ_2 and δ_3 in order to fulfill the constraints $\delta_i \geq 0$ and $\sum_{i=1}^{3} \delta_i = 1$.

3.3. *Selection of bases for projecting: A regularised gradient-based technique using CKTA*

The above mentioned methodology is not suitable for choosing the optimal set of bases for projecting the data. Therefore, once the kernel matrix has been optimized by the process presented in the previous subsection, we now present a regularised gradient ascent methodology to improve the alignment of the kernel matrix by selecting some of its bases.

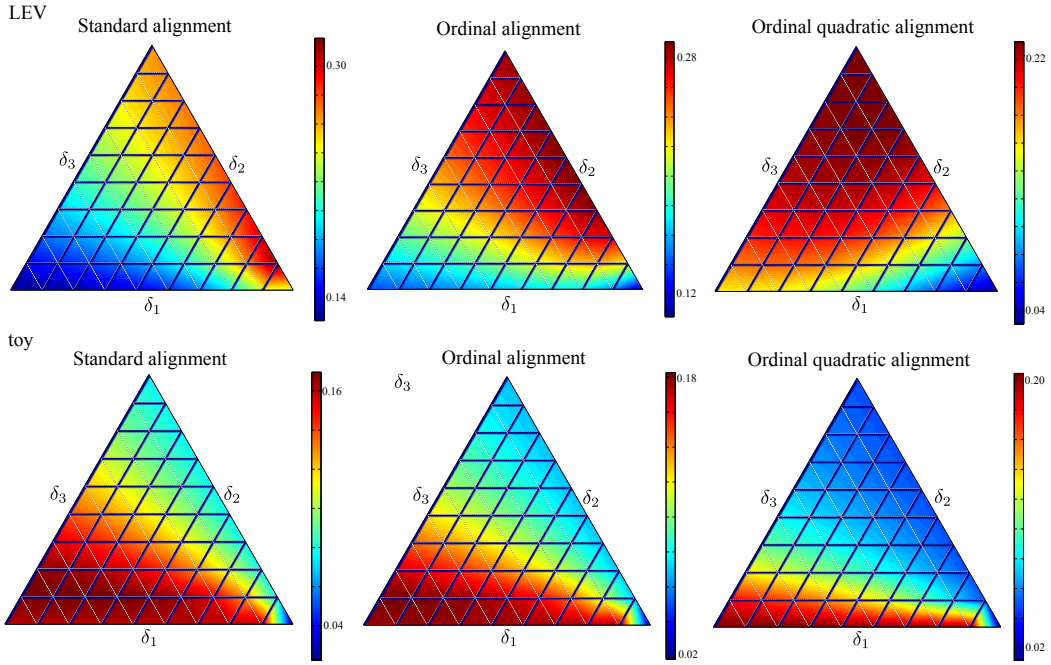


Figure 4: Kernel-target alignment optimisation surfaces for $\{\delta_1, \delta_2, \delta_3\}$ for the LEV and toy datasets and three different weighting matrices (see Table 2).

Usually, the j dominant eigenvectors (the ones associated to the highest eigenvalues) are used as a projection onto a subspace to remove noise (as done in principal components analysis) or for visualisation purposes. Therefore, the eigenvalues ranking from $j+1$ to r (and the corresponding eigenvectors) are discarded so that $\mathbf{K}_j = \sum_{i=1}^{j} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$. By using this idea, the distance to the original kernel r-rank matrix $\mathbf{K}$ (*i.e.*, $\|\mathbf{K} - \mathbf{K}_j\|_{\mathrm{F}}^2$) is minimised over all rank-j matrices. However, in this case, we aim to find the projection that minimises $\|\mathbf{K}^* - \mathbf{K}_j\|_{\mathrm{F}}^2$ for $\mathbf{K}^*$ being the ideal kernel. Note that, since $\mathbf{K}$ does not include any information about the target variable, the bases associated to the highest eigenvalues of $\mathbf{K}$ do not have to be so informative. Alternatively, we can form a matrix:

$$\mathbf{K}_{\mathbf{w}} = \sum_{i=1}^{r} f(w_i) \lambda_i \mathbf{u}_i \mathbf{u}_i^T, \tag{10}$$

where $f(w_i) \in [0, 1]$ so as to maintain $\mathbf{K}_{\mathbf{w}}$ to be PSD and for simplicity. In this way, the weight of the eigenvectors is now determined by $f(w_i)$ and λ_i. The objective of this definition is to generalise the combination of the eigenvectors in order to obtain information about which of them are more important for improving the CKTA. We aim to find a lower subspace for our data that maintains the labelling information in a proper way. We propose to define the optimisation problem as follows:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \left(\mathcal{A}_c(\mathbf{K}_{\mathbf{w}}, \mathbf{K}^*) - \frac{\mu}{r} \sum_{i=1}^{r} f(w_i) \right), \tag{11}$$

where $f(w_i) \in [0, 1]$ and μ is a regularisation parameter. L^1 or L^2 norms could be considered for the weights (*i.e.*, $f(w_i) = |w_i|$ or $f(w_i) = w_i^2$, respectively). For simplicity, we choose:

$$f(w_i) = \frac{1}{1 + e^{-w_i}}, \tag{12}$$

i.e., the sigmoid function. We experimentally found that this formulation promotes more sparsity than L^2 norm, while being still derivable. As we apply gradient descent optimisation for optimisation, considering L^1-norm would imply a constrained problem (with a higher computational cost) or applying iterative techniques similar to those in [23].

Because of the differentiability of the function to maximise in Eq. (11) (which will be named g from now on) with respect to the vector $\mathbf{w}$, a gradient ascent algorithm can be used to maximise it. The gradient vector will be composed of the following partial derivatives $\nabla g = \left[\frac{\partial g}{\partial w_1}, \ldots, \frac{\partial g}{\partial w_r}\right]^{\mathrm{T}}$. The iRprop$^+$ algorithm is considered for optimising the aforementioned function, because of its proven robustness for optimising KTA [24]. Each parameter w_i will be updated considering the sign of $\frac{\partial g}{\partial w_i}$ but not the magnitude. Although the second partial derivatives can also be computed and used for optimisation, they actually make the process more computationally costly due to the complexity of their formulae.

The first derivative of g with respect to w_i is:

$$\frac{\partial g}{\partial w_i} = \frac{\partial \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\mathbf{w}}, \mathbf{K}^*)}{\partial w_i} - \frac{\mu}{r} \cdot \frac{\partial f}{\partial w_i}, \tag{13}$$

where the alignment derivative with respect to w_i is:

$$\frac{\partial \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\mathbf{w}}, \mathbf{K}^*)}{\partial w_i} = \tag{14}$$

$$= \frac{1}{\|\mathbf{K}_{\mathrm{c}}^*\|_{\mathrm{F}}} \left[\frac{\left\langle \frac{\partial \mathbf{K}_{\mathbf{w}}}{\partial w_i}, \mathbf{K}_{\mathrm{c}}^* \right\rangle_{\mathrm{F}}}{\|\mathbf{K}_{\mathbf{w}_{\mathrm{c}}}\|_{\mathrm{F}}} - \frac{\langle \mathbf{K}_{\mathbf{w}}, \mathbf{K}_{\mathrm{c}}^* \rangle_{\mathrm{F}} \cdot \left\langle \mathbf{K}_{\mathbf{w}_{\mathrm{c}}}, \frac{\partial \mathbf{K}_{\mathbf{w}}}{\partial w_i} \right\rangle_{\mathrm{F}}}{\|\mathbf{K}_{\mathbf{w}_{\mathrm{c}}}\|_{\mathrm{F}}^3} \right], \tag{15}$$

and, for matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, it is satisfied that $\langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_1, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_2 \rangle_{\mathrm{F}}$ [12], which simplifies the computation. The computation of the KTA takes $O(m^2)$ operations per

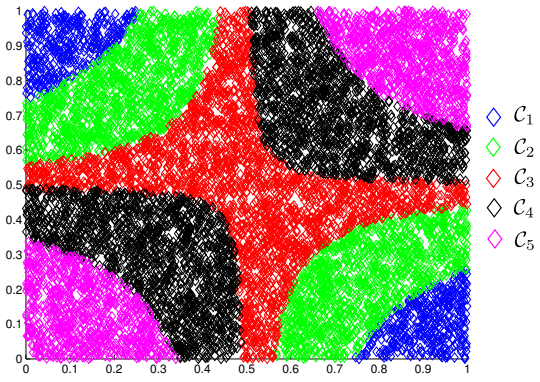


Figure 5: Two-dimensional representation of the structure of an ordinal nonlinearly separable toy dataset.

parameter w_i to optimise [25]. Because this optimisation does not involve any additional optimisation problem, it is very fast in practice. The derivative of the kernel (see Eq. (10)) is in this case:

$$\frac{\partial \mathbf{K_w}}{\partial w_i} = \frac{\partial f}{\partial w_i}\lambda_i \mathbf{u}_i \mathbf{u}_i^T = \frac{e^{-w_i}}{(1+e^{-w_i})^2}\lambda_i \mathbf{u}_i \mathbf{u}_i^T. \quad (16)$$

The regularisation term in Eq. (11) (*i.e.*, the term $\mu \sum_{i=1}^{r} f(w_i)$) results in the less representative bases presenting a w_i value close to zero, and the most representative ones close to one. The bases with w_i close to one are the ones chosen for projecting the data. The parameter μ is set by cross-validation. After applying the gradient ascent method, those bases which $f(w_i) < 10^{-6}$ are eliminated from the kernel matrix, and the original eigenvalues of the remaining bases are taken into account to reconstruct the matrix.

Although we have considered original CKTA for all these definitions, ordinal CKTA can be similarly considered by applying the weight matrix to the ideal kernel matrix. If we include the ordinal version of CKTA in Eq. (11), the projections that better maintain the ordinal similarity information will be preferred.

The convergence of the proposal to a proper solution is now discussed. Consider the case when there is a basis that perfectly projects the patterns according to the labelling (for example, the eigenvalue λ_1 and the associated eigenvector $\mathbf{u}_1$). Let denote the projected kernel matrix using this basis as $\mathbf{K}_{\lambda_1} = f(w_1)\lambda_1 \mathbf{u}_1 \mathbf{u}_1^T$. Furthermore, consider other basis (λ_2, $\mathbf{u}_2$ and $\mathbf{K}_{\lambda_2}$) which projection results in no useful information about the labelling. That is, $\mathcal{A}_c(\mathbf{K}_{\lambda_1}, \mathbf{K}^*) = 1$ (or, in other words $\mathbf{K}_{\lambda_1} = c\mathbf{K}^*$, where c is a scalar) and $\mathcal{A}_c(\mathbf{K}_{\lambda_2}, \mathbf{K}^*) = 0$. Let suppose $r = 2$, so that $\mathbf{K_w} = \mathbf{K}_{\lambda_1} + \mathbf{K}_{\lambda_2}$. Under these assumptions, what we would like to show is how $f(w_2)$ influences $\mathcal{A}_c(\mathbf{K_w}, \mathbf{K}^*)$. First, the alignment can also be defined as:

$$\mathcal{A}_c(\mathbf{K_w}, \mathbf{K}^*) = \frac{\mathrm{Tr}(\mathbf{K_w}\mathbf{K}_c^*)}{\sqrt{\mathrm{Tr}((\mathbf{K}_c^*)^2)\mathrm{Tr}(\mathbf{K}_{\mathbf{wc}}^2)}} = \frac{\mathrm{Tr}(\mathbf{K}_{\lambda_1}\mathbf{K}_c^*) + \mathrm{Tr}(\mathbf{K}_{\lambda_2}\mathbf{K}_c^*)}{\sqrt{\mathrm{Tr}((\mathbf{K}_c^*)^2)\mathrm{Tr}(\mathbf{K}_{\mathbf{wc}}^2)}}. \quad (17)$$

Note that the fact that $\mathcal{A}_c(\mathbf{K}_{\lambda_2}, \mathbf{K}^*) = 0$ comes from $\mathrm{Tr}(\mathbf{K}_{\lambda_2}\mathbf{K}_c^*) = 0$. Besides, $\mathrm{Tr}(\mathbf{K}_{\mathbf{wc}}^2) = (f(w_1)\lambda_1)^2 + (f(w_2)\lambda_2)^2$. Note that the only case in which $\mathcal{A}_c(\mathbf{K_w}, \mathbf{K}^*) = \mathcal{A}_c(\mathbf{K}_{\lambda_1}, \mathbf{K}^*)$ (*i.e.*, the maximum value) is for $f(w_2) = 0$, which makes $\mathrm{Tr}(\mathbf{K_w}\mathbf{K}_c^*) = \mathrm{Tr}(\mathbf{K}_{\mathbf{wc}}^2)$.

Moreover, given the definition of $\mathbf{K}_{\lambda_1}$ and $\mathbf{K}_{\lambda_2}$ and the fact that $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthonormal, *i.e.*, $\mathrm{Tr}(\mathbf{K}_{\lambda_1 \mathrm{c}}\mathbf{K}_{\lambda_2 \mathrm{c}}) = 0$, the alignment between these two matrices is zero (they are uncorrelated). In this way, the alignment provided by one basis does not affect the alignment provided by the others.

Finally, note that this projection technique can also be used for visualisation purposes in supervised learning contexts (as an analogue technique for Kernel Principal Component Analysis for non-supervised problems), by optimising the bases and then representing the projection onto the two or three most dominant bases.

Table 3: Characteristics of the 28 benchmark datasets used for the experiments.

Dataset	#Pat.	#Attr.	#Classes	Class distribution
contact-lenses	24	6	3	(15,5,4)
pasture	36	25	3	(12,12,12)
squash-stored	52	51	3	(23,21,8)
squash-unstored	52	52	3	(24,24,4)
tae	151	54	3	(49, 50, 52)
SWD	1000	10	4	(32,352,399,217)
car	1728	21	4	(1210,384,69,65)
diabetes5	43	2	5	(5,6,22,8,2)
pyrim5	74	27	5	(7,28,17,12,10)
triazines5	186	60	5	(7,10,26,86,57)
wisconsin5	194	32	5	(67,41,43,24,19)
machine5	209	6	5	(152,27,13,7,10)
toy	300	2	5	(35,87,79,68,31)
auto5	392	7	5	(91,131,101,59,10)
housing5	506	13	5	(77,239,123,36,31)
eucalyptus	736	91	5	(180, 107, 130, 214, 105)
stock5	950	9	5	(158,227,272,207,86)
LEV	1000	4	5	(93,280,403,197,27)
automobile	205	71	6	(3,22,67,54,32,27)
heating	768	8	8	(20,265,112,51,119,85,82,34)
cooling	768	8	8	(150,198,52,114,126,89,26,13)
diabetes10	43	2	10	(2,3,3,3,10,12,4,2,2,2)
pyrim10	74	27	10	(2,2,14,14,13,5,10,4,3,7)
triazines10	186	60	10	(4,3,2,8,11,15,36,50,45,12)
wisconsin10	194	32	10	(46,21,28,13,25,18,14,10,9,10)
machine10	209	6	10	(115,37,21,6,8,5,3,4,4,6)
auto10	392	7	10	(13,78,73,58,53,48,37,22,4,6)
housing10	506	13	10	(22,55,85,154,84,39,29,7,10,21)
stock10	950	9	10	(48,110,108,119,168,104,104,103,64,22)

All nominal variables are transformed into binary ones.
For discretised datasets, the number included in their names (5 or 10) represents the number of bins considered during discretisation.

4. Experimental results

This section presents the experimental part of the paper: the datasets and methods tested, the evaluation measures and, finally, the results obtained.

4.1. Datasets

Several benchmark datasets have been considered in order to validate the methodologies proposed; some publicly available real ordinal classification datasets were extracted from the UCI and `mldata.org` repositories [26, 27] and some of the ordinal regression benchmark datasets provided by Chu et. al [28] were considered due to their widespread use in ordinal regression [5, 29]. The latter do not originally represent ordinal

classification tasks but regression ones, where the target variable is discretised into Q different bins (representing classes) with equal bining, to turn regression into ordinal classification. Table 3 presents the main characteristics of 28 the datasets used for the experiments. A synthetic two-dimensional toy dataset has been included in the experiments. The representation of this dataset can be seen in Figure 5.

Regarding the experimental setup, a 30-holdout stratified technique has been applied to divide the datasets, using 75% of the patterns for training the model, and the remaining 25% for testing it. One model is obtained and evaluated for each split. Finally, the results are taken as the mean and standard deviation over each one of the test sets.

4.2. Metrics considered

Concerning evaluation measures, several metrics can be considered for ordinal classifiers, but the most common ones in machine learning are the Mean Absolute Error (MAE) and the Accuracy (Acc) [2, 5, 29], where the MAE is the average deviation in absolute value of the predicted class from the true class [21], $MAE = (1/N)\sum_{i=1}^{N} e(\mathbf{x}_i)$, where $e(x_i) = |r(y_i) - r(y_i^*)|$ is the distance between the true and the predicted ranks. MAE values range from 0 to $Q-1$ (maximum deviation in number of ranks between two labels). Instead, Acc penalises all mistakes equally.

4.3. Methods tested

To test the different proposals in Section 3, we consider the comparison of the following methods:

- The POM algorithm in the original input space $\mathcal{X}$, which is a linear method (POM).
- A kernelisation of the POM algorithm, cross-validating the number of dimensions for the projected subspace and the width of the Gaussian kernel (K-POM). The dimensions selected are always those with the highest eigenvalues. For this and the following three methods, the EFS was considered to perform this kernelisation, as introduced in Section 3.1.
- The POM algorithm kernelised using a regularised gradient-based technique for selecting the dominant dimensions (KRGB-POM). The width of the Gaussian kernel is also selected through cross-validation, so the difference between KRGB-POM and K-POM lies only on the selection of the dominant dimensions through the regularised gradient-based technique presented in Section 3.3.
- Kernelised version of the POM algorithm solving a QP optimisation problem for learning the kernel presented in Section 3.2.1 (instead of cross-validation), and the regularised gradient-based technique for selecting the dominant dimensions of Section 3.3 (KLRGB-POM). The original version of CKTA was considered.
- Finally, we also tested the KLRGB-POM methodology described above, but considering the notion of ordinal CKTA (introduced in Section 3.2) for both kernel optimisations (OKLRGB-POM).

The source code in Matlab for the proposed methods can be downloaded from the web associated to this paper[1].

Furthermore, two well-known kernel methods for ordinal regression have been chosen for comparison purposes (Kernel Discriminant Learning for Ordinal Regression [5], KDLOR, and Support Vector for Ordinal Regression with Implicit Constraints [29], SVORIM).

For model selection, a stratified nested 3-fold cross-validation has been applied to the training sets, with kernel width within the values $\{10^{-2}, 10^{0}, 10^{2}\}$. The same values are considered for the cost parameter of SVORIM. The cross-validation criterion is the MAE, since it can be considered the most common one in ordinal regression. The kernel selected for all the algorithms is the Gaussian one, $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2}\right)$ where σ is the width of the kernel. The logit function has been used for all the POM-based algorithms. The number of dimensions for the empirical feature space (j) has been cross-validated within the values $\{10, 20, 30\}$. For KLRGB-POM and OKLRGB-POM, the number of kernels and widths considered (p) are the same than those used for cross-validation ($\{10^{-2}, 10^{0}, 10^{2}\}$).

Concerning the gradient-based technique, the initial points for all the methods tested were randomly chosen from a uniform distribution $U[0, 1]$. The gradient norm stopping criterion was set at 10^{-5} and the maximum number of conjugate gradient steps at 50. Furthermore, the μ parameter associated to the regularisation was cross-validated within the values $\{10^{-4}, 10^{-2}, 10^{0}\}$.

4.4. Results

The results of the battery of experiments can be seen in Table 4 (for Acc) and Table 5 (for MAE), where all the methods described in the previous subsection have been tested. To better summarise these results, these tables also show the test mean rankings in terms of Acc and MAE for all the methods considered in this experiment, along all of the 28 datasets. For each dataset, a ranking of 1 is given to the best method in average, and a 7 is given to the worst one. It can be seen, that simply cross-validating the number of dimensions, the POM algorithm can be improved to a great extent (POM versus K-POM comparison), and that the use of a more intelligent technique for selecting the bases for projecting is a good option (KRGB-POM versus K-POM). The QP kernel learning technique of methods KLRGB-POM and OKLRGB-POM also improve the results. Finally, it can be seen that using a weighting matrix in CKTA can help to improve the results in terms of MAE (KLRGB-POM versus OKLRGB-POM). The results also show that the proposals are competitive with the selected ordinal state-of-the-art methods (KDLOR and SVORIM) and are able to outperform the standard linear POM algorithm in most cases. The cases of the toy and eucalyptus datasets are very good examples of the

[1] `http://www.uco.es/grupos/ayrna/neucom-kpom`

Table 4: Results obtained for each method reported in terms of *Acc*.

Dataset	POM	K-POM	KRGB-POM	KLRGB-POM	OKLRGB-POM	KDLOR	SVORIM
contact-lenses	**65.56 ± 15.74**	50.00 ± 20.53	60.00 ± 18.88	62.23 ± 13.79	62.22 ± 13.79	48.89 ± 21.41	*63.89 ± 8.84*
pasture	46.30 ± 13.40	60.74 ± 16.18	63.33 ± 13.42	64.44 ± 13.18	*65.19 ± 11.57*	60.74 ± 11.20	**66.67 ± 11.30**
squash-stored	38.97 ± 15.38	60.51 ± 12.73	**69.49 ± 12.20**	*66.15 ± 11.36*	65.13 ± 11.02	65.38 ± 13.36	63.33 ± 11.37
squash-unstored	36.67 ± 14.66	62.05 ± 14.27	76.15 ± 11.13	**77.95 ± 12.08**	*76.92 ± 12.29*	73.33 ± 13.81	75.38 ± 12.02
tae	43.86 ± 11.49	36.32 ± 6.00	57.28 ± 7.17	**57.89 ± 6.37**	*57.46 ± 6.26*	57.28 ± 5.43	57.19 ± 6.87
SWD	52.88 ± 3.22	55.81 ± 2.84	*57.28 ± 3.38*	**57.51 ± 3.45**	57.09 ± 3.61	47.93 ± 2.98	56.73 ± 2.78
diabetes5	42.73 ± 10.98	41.52 ± 14.46	44.55 ± 13.58	48.18 ± 8.32	*49.39 ± 7.80*	**52.42 ± 5.69**	49.09 ± 7.40
pyrim5	46.67 ± 10.95	*59.90 ± 8.13*	57.37 ± 9.23	57.72 ± 8.67	**60.00 ± 9.11**	49.12 ± 10.82	58.77 ± 9.88
triazines5	30.85 ± 1.08	45.39 ± 3.84	42.70 ± 10.04	44.40 ± 5.05	44.82 ± 4.36	**46.60 ± 2.46**	*45.96 ± 2.66*
wisconsin5	*28.50 ± 6.33*	**29.52 ± 4.24**	24.76 ± 8.11	20.95 ± 3.51	20.75 ± 3.64	21.22 ± 1.48	26.33 ± 4.89
machine5	**85.16 ± 4.28**	83.84 ± 5.31	83.33 ± 5.03	82.83 ± 5.50	*83.90 ± 4.02*	82.14 ± 4.40	83.58 ± 3.87
toy	30.13 ± 5.36	91.15 ± 3.44	*92.09 ± 3.05*	91.16 ± 3.14	90.84 ± 3.60	88.93 ± 3.17	**94.76 ± 2.57**
auto5	62.11 ± 5.04	**75.82 ± 3.77**	*75.37 ± 4.15*	74.32 ± 4.60	74.08 ± 4.92	70.99 ± 4.55	74.76 ± 3.37
housing5	60.68 ± 4.09	73.12 ± 4.08	73.81 ± 4.05	*76.17 ± 2.88*	**76.98 ± 2.80**	73.81 ± 3.14	75.51 ± 3.17
eucalyptus	15.02 ± 1.51	52.92 ± 3.47	55.60 ± 3.17	**61.54 ± 2.43**	*61.07 ± 5.56*	56.90 ± 3.69	60.69 ± 2.58
stock5	63.77 ± 2.25	84.24 ± 1.86	87.59 ± 1.67	*88.98 ± 1.68*	**89.02 ± 1.98**	85.28 ± 2.05	87.87 ± 1.83
LEV	53.92 ± 3.18	61.95 ± 2.69	62.15 ± 2.41	*62.40 ± 2.77*	**62.60 ± 2.81**	54.60 ± 2.88	61.72 ± 2.87
automobile	38.78 ± 20.14	53.59 ± 5.82	69.62 ± 6.69	64.87 ± 4.89	65.90 ± 5.39	**70.71 ± 6.32**	*70.71 ± 4.19*
heating	53.02 ± 3.80	69.41 ± 2.70	**86.77 ± 2.89**	81.25 ± 3.78	*82.41 ± 1.91*	78.23 ± 6.64	74.74 ± 5.43
cooling	50.36 ± 3.84	64.84 ± 2.41	**73.63 ± 3.01**	71.04 ± 2.49	70.82 ± 2.65	*72.64 ± 3.49*	68.61 ± 5.02
diabetes10	**25.15 ± 11.62**	20.30 ± 9.75	*23.03 ± 8.85*	22.73 ± 7.83	*23.03 ± 8.85*	19.09 ± 8.04	20.00 ± 10.78
pyrim10	32.81 ± 10.41	*34.21 ± 10.95*	30.18 ± 11.80	22.64 ± 5.01	22.63 ± 5.55	30.18 ± 8.73	**37.72 ± 7.58**
triazines10	6.23 ± 0.80	29.29 ± 4.73	24.18 ± 10.42	**30.35 ± 6.34**	29.15 ± 7.56	23.76 ± 4.26	*29.86 ± 3.28*
wisconsin10	**15.92 ± 5.48**	*12.65 ± 4.37*	12.45 ± 5.98	10.82 ± 2.99	10.61 ± 3.69	6.53 ± 0.83	11.22 ± 3.93
machine10	64.84 ± 6.51	*67.99 ± 4.87*	67.36 ± 6.15	**68.43 ± 5.43**	66.98 ± 4.14	65.03 ± 5.28	65.41 ± 5.11
auto10	37.24 ± 4.53	*55.71 ± 4.88*	55.10 ± 4.99	51.36 ± 11.00	52.72 ± 10.56	47.31 ± 4.86	**56.12 ± 4.48**
housing10	35.96 ± 2.96	58.01 ± 5.18	*58.32 ± 3.93*	56.33 ± 4.96	56.51 ± 4.02	55.30 ± 4.23	**59.00 ± 3.97**
stock10	34.10 ± 3.02	68.45 ± 2.42	74.85 ± 3.21	**82.31 ± 2.13**	*82.28 ± 2.10*	71.76 ± 3.57	79.08 ± 2.37
Ranking	5.71	4.46	3.36	**3.04**	3.29	4.95	*3.20*
Friedman's test: Confidence interval $C_0 = (0, F_{\alpha=0.05}) = 2.15$, F-val$_{Acc}$: $8.24 \notin C_0$							

The best method is in **bold** face and the second one in *italics*.

Table 5: Results obtained for each method reported in terms of *MAE*.

Dataset	POM	K-POM	KRGB-POM	KLRGB-POM	OKLRGB-POM	KDLOR	SVORIM
contact-lenses	0.500 ± 0.255	0.722 ± 0.298	0.561 ± 0.257	*0.378 ± 0.138*	**0.377 ± 0.138**	0.656 ± 0.239	0.522 ± 0.122
pasture	0.589 ± 0.168	0.426 ± 0.168	0.370 ± 0.132	0.356 ± 0.132	*0.348 ± 0.116*	0.396 ± 0.116	**0.333 ± 0.113**
squash-stored	0.792 ± 0.260	0.426 ± 0.153	**0.308 ± 0.128**	*0.344 ± 0.121*	0.351 ± 0.117	0.362 ± 0.147	0.377 ± 0.118
squash-unstored	0.797 ± 0.246	0.385 ± 0.146	0.238 ± 0.111	**0.221 ± 0.121**	*0.231 ± 0.123*	0.267 ± 0.138	0.246 ± 0.120
tae	0.751 ± 0.200	0.650 ± 0.063	0.533 ± 0.106	*0.456 ± 0.065*	0.465 ± 0.067	**0.453 ± 0.058**	0.468 ± 0.071
SWD	0.504 ± 0.035	0.460 ± 0.029	*0.446 ± 0.037*	**0.445 ± 0.035**	0.448 ± 0.037	0.591 ± 0.033	*0.446 ± 0.029*
diabetes5	0.670 ± 0.130	0.733 ± 0.151	0.718 ± 0.272	0.633 ± 0.179	**0.620 ± 0.143**	*0.621 ± 0.093*	0.667 ± 0.099
pyrim5	0.711 ± 0.155	**0.442 ± 0.114**	0.474 ± 0.129	0.470 ± 0.110	0.465 ± 0.115	0.596 ± 0.124	*0.449 ± 0.125*
triazines5	1.053 ± 0.032	*0.677 ± 0.049*	0.842 ± 0.428	0.738 ± 0.131	0.713 ± 0.080	**0.671 ± 0.032**	*0.677 ± 0.035*
wisconsin5	1.144 ± 0.156	**1.007 ± 0.088**	1.401 ± 0.373	1.041 ± 0.051	1.043 ± 0.051	1.110 ± 0.022	*1.040 ± 0.058*
machine5	*0.178 ± 0.045*	0.198 ± 0.052	0.195 ± 0.064	0.184 ± 0.063	**0.177 ± 0.043**	0.214 ± 0.062	0.181 ± 0.038
toy	0.944 ± 0.122	0.090 ± 0.034	*0.079 ± 0.031*	0.088 ± 0.031	0.092 ± 0.036	0.111 ± 0.032	**0.052 ± 0.026**
auto5	0.385 ± 0.054	**0.246 ± 0.040**	*0.251 ± 0.043*	0.255 ± 0.050	0.265 ± 0.053	0.297 ± 0.051	0.262 ± 0.036
housing5	0.404 ± 0.041	0.283 ± 0.045	0.270 ± 0.045	*0.250 ± 0.031*	**0.243 ± 0.033**	0.269 ± 0.032	0.251 ± 0.034
eucalyptus	1.940 ± 0.276	0.557 ± 0.045	0.529 ± 0.051	**0.436 ± 0.031**	0.453 ± 0.142	0.504 ± 0.046	*0.439 ± 0.032*
stock5	0.375 ± 0.022	0.158 ± 0.019	0.124 ± 0.017	*0.111 ± 0.017*	**0.110 ± 0.020**	0.148 ± 0.021	0.121 ± 0.018
LEV	0.505 ± 0.033	0.418 ± 0.029	0.416 ± 0.024	*0.413 ± 0.031*	**0.412 ± 0.030**	0.514 ± 0.034	0.420 ± 0.030
automobile	1.153 ± 0.750	0.522 ± 0.072	0.411 ± 0.101	0.402 ± 0.073	0.393 ± 0.076	*0.384 ± 0.088*	**0.368 ± 0.075**
heating	0.555 ± 0.040	0.341 ± 0.032	**0.134 ± 0.028**	0.200 ± 0.058	*0.184 ± 0.022*	0.225 ± 0.067	0.273 ± 0.065
cooling	0.580 ± 0.049	0.396 ± 0.026	**0.272 ± 0.031**	0.307 ± 0.028	0.305 ± 0.030	*0.296 ± 0.036*	0.350 ± 0.062
diabetes10	1.442 ± 0.331	1.645 ± 0.296	1.500 ± 0.351	*1.382 ± 0.328*	**1.352 ± 0.222**	1.521 ± 0.256	1.527 ± 0.291
pyrim10	1.344 ± 0.214	*1.058 ± 0.196*	1.351 ± 0.429	1.379 ± 0.163	1.332 ± 0.193	1.342 ± 0.274	**0.995 ± 0.185**
triazines10	2.742 ± 0.417	*1.311 ± 0.095*	2.188 ± 1.363	1.409 ± 0.426	1.548 ± 0.654	1.438 ± 0.081	**1.288 ± 0.095**
wisconsin10	2.431 ± 0.190	**2.224 ± 0.139**	3.228 ± 0.780	2.251 ± 0.124	*2.232 ± 0.087*	2.359 ± 0.051	2.319 ± 0.099
machine10	0.534 ± 0.137	0.501 ± 0.132	0.516 ± 0.142	**0.451 ± 0.099**	*0.464 ± 0.081*	0.531 ± 0.146	0.482 ± 0.109
auto10	0.769 ± 0.070	*0.518 ± 0.058*	0.525 ± 0.067	0.645 ± 0.412	0.610 ± 0.399	0.680 ± 0.075	**0.504 ± 0.057**
housing10	0.844 ± 0.061	0.525 ± 0.073	*0.502 ± 0.058*	0.562 ± 0.099	0.580 ± 0.074	0.521 ± 0.056	**0.482 ± 0.059**
stock10	0.870 ± 0.049	0.319 ± 0.025	0.258 ± 0.032	*0.180 ± 0.021*	**0.179 ± 0.021**	0.290 ± 0.035	0.212 ± 0.024
Ranking	6.25	4.52	4.09	2.93	**2.71**	4.54	*2.96*
Friedman's test: Confidence interval $C_0 = (0, F_{\alpha=0.05}) = 2.15$, F-val$_{Acc}$: $13.86 \notin C_0$							

The best method is in **bold** face and the second one in *italics*.

capability of the proposals to deal with nonlinearly separable data. Indeed, in those datasets where the POM has achieved better results, the proposed methods also obtained a similar performance.

As can be observed in the results, OKLRGB-POM performs better than KLRGB-POM in *MAE* but worse in *Acc*. This is a consequence of the cost matrices introduced in Section 3.2, where a uniform cost for all errors (KLRGB) is clearly favour-

ing accuracy, while non-uniform costs (OKLRGB) are better for reducing *MAE*. However, when dealing with ordinal regression problems, classifiers obtaining better *MAE* results are generally preferred. Similar conclusions were found in [29] when comparing the results obtained by SVORIM to its explicit version, SVOREX (where only adjacent classes are taken into account for the slacks).

To determine the statistical significance of the differences observed between the different methodologies, a procedure to compare multiple classifiers in multiple datasets is employed [30]. Table 4 and Table 5 also show the result of applying the statistical non-parametric Friedman's test (for a significance level of and $\alpha = 0.05$) to the mean *Acc* and *MAE* rankings. It can be seen that the test rejects the null-hypothesis that all of the algorithms perform similarly in mean ranking for all the metrics (note that for *MAE* the significant differences are larger).

On the basis of this rejection and following the guidelines in [30], we consider the best performing methods in *Acc* and *MAE* (*i.e.*, KLRGB-POM and OKLRGB-POM, respectively) as control methods for the following tests. Furthermore, we also consider the method POM as a control method, to analyse the performance of the original linear method with respect to the rest of developed techniques. We compare these three methods to the rest according to their rankings. The Holm's test is an approach to compare all classifiers to a given classifier (a control method). The test statistics for comparing the i-th and j-th method using this procedure is:

$$z = \frac{R_i - R_l}{\sqrt{\frac{L(L+1)}{6T}}},$$

where L is the number of algorithms, T is the number of datasets and R_i is the mean ranking of the i-th method. The z value is used to find the corresponding probability from the table of the normal distribution, which is then compared with an appropriate level of significance α. Holm's test adjusts the value for α in order to compensate for multiple comparisons. This is done in a step-up procedure that sequentially tests the hypotheses ordered by their significance. We denote the ordered p-values by $p_1, p_2, \ldots, p_q$ so that $p_1 \leq p_2 \leq \ldots \leq p_q$. Holm's test compares each p_i with $\alpha^*_{\text{Holm}} = \alpha/(L-i)$, starting from the most significant p value. If p_1 is below $\alpha/(L-1)$, the corresponding hypothesis is rejected and we allow to compare p_2 with $\alpha/(L-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on.

This process is included in Table 6, where the results from the Holm statistical test are shown. Several conclusions can be drawn. First, it can be seen that the POM algorithm is significantly improved by most of the algorithms in terms of *Acc* and *MAE*, specially by KLRGB-POM in *Acc* and OKLRGB-POM in *MAE*. Considering the KLRGB-POM method, one can appreciate that it significantly outperforms the K-POM technique, meaning this that the use of the regularised gradient-based technique for selecting the optimal dimensions helps to improve the performance of the proposed kernelisation of the POM method. It is also better than the KDLOR method. However, no significant differences can be seen between this technique and the rest of methodologies that make use of this regularised gradient-based technique (although it presents better performance in mean ranking, as can be seen in Table 4 and Table 5). Concerning the ordinal version (OKLRGB-POM), similar results can be found, although in this case, there exists significant differences with respect to KRGB-POM in *MAE* (which is similar to KLRGB-POM but using gradient descent for adjusting the kernel). As can be seen, the developed techniques present statistically significant differences when compared to KDLOR, and improved SVORIM results (although not significantly). We should take into account that SVORIM is indeed one of the most successful and widely used technique in the state-of-the-art of ordinal regression [2]. As a summary, both multiple kernel proposals (OKLRGB-POM and KLRGB-POM) improve the results of POM and other kernel techniques (KDLOR and K-POM), while OKLRGB-POM is also able to improve the results from the proposal based on gradient descent (KRGB-POM). The kernelisation strategy is suitable for enabling the POM method to perform nonlinear decision boundaries and to reach the state-of-the-art results (SVORIM), while still obtaining natural probability estimations, which can only be approximated by POM.

We now analyse how the selection of the dimensions differ for all the datasets. This is done by considering K-POM and KRGB-POM methods, that make use of different strategies for selecting the dominant dimensions of the projected subspace and result in very different performance. Table 7 reports the percentage of agreement between both selections (i.e, if both algorithms consider the base $\mathbf{u}_i$ to be suitable, or the contrary). From this result, it can be seen that although from certain datasets the level of agreement is very high (meaning this that the selected dimensions for the KRGB-POM method are the ones associ-

Table 6: Results of the Holm procedure using POM, KLRGB-POM and OKLRGB-POM as control methods: corrected α values, compared method and p-values, ordered by the number of comparison (i).

Control alg.: POM			*Acc*		*MAE*	
i	$\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1	0.0083	0.0167	KLRGB-POM	0.0000_{--}	OKLRGB-POM	0.0000_{--}
2	0.0100	0.0200	SVORIM	0.0000_{--}	KLRGB-POM	0.0000_{--}
3	0.0125	0.0250	OKLRGB-POM	0.0000_{--}	SVORIM	0.0000_{--}
4	0.0167	0.0333	KRGB-POM	0.0000_{--}	KRGB-POM	0.0001_{--}
5	0.0250	0.0500	K-POM	0.0304_{-}	K-POM	0.0027_{--}
6	0.0500	0.1000	KDLOR	0.1853	KDLOR	0.0029_{--}
Control alg.: KLRGB-POM			*Acc*		*MAE*	
i	$\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1	0.0083	0.0167	POM	0.0000_{++}	POM	0.0000_{++}
2	0.0100	0.0200	KDLOR	0.0009_{++}	KDLOR	0.0053_{++}
3	0.0125	0.0250	K-POM	0.0133_{+}	K-POM	0.0059_{++}
4	0.0167	0.0333	KRGB-POM	0.5777	KRGB-POM	0.0444
5	0.0250	0.0500	OKLRGB-POM	0.6650	OKLRGB-POM	0.7105
6	0.0500	0.1000	SVORIM	0.7807	SVORIM	0.9507
Control alg.: OKLRGB-POM			*Acc*		*MAE*	
i	$\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1	0.0083	0.0167	POM	0.0000_{++}	POM	0.0000_{++}
2	0.0100	0.0200	KDLOR	0.0040_{++}	KDLOR	0.0016_{++}
3	0.0125	0.0250	K-POM	0.0412	K-POM	0.0018_{++}
4	0.0167	0.0333	KLRGB-POM	0.6650	KRGB-POM	0.0172_{+}
5	0.0250	0.0500	SVORIM	0.8771	SVORIM	0.6650
6	0.0500	0.1000	KRGB-POM	0.9015	KLRGB-POM	0.7105

Win (++) or lose (−−) with statistical significant difference for $\alpha = 0.05$
Win (+) or lose (−) with statistical difference with $\alpha = 0.10$

ated to the first eigenvalues), for most of the datasets, the level of agreement is medium or low, indicating therefore that the selection of the most suitable dimensions is necessary.

Table 7: Agreement between the selected dimensions for K-POM and KRGB-POM methods.

Dataset	Mean ± Std	Dataset	Mean ± Std
contact-lenses	55.19 ± 21.49	eucalyptus	31.15 ± 10.16
pasture	64.07 ± 14.82	stock5	68.15 ± 16.80
squash-stored	65.81 ± 21.83	LEV	84.67 ± 14.49
squash-unstored	68.63 ± 18.57	automobile	30.57 ± 7.39
tae	40.62 ± 11.88	heating	74.84 ± 6.65
SWD	83.37 ± 12.12	cooling	77.32 ± 8.39
diabetes5	79.68 ± 17.22	diabetes10	74.77 ± 16.26
pyrim5	47.94 ± 19.93	pyrim10	39.80 ± 13.90
triazines5	19.51 ± 7.51	triazines10	22.89 ± 11.10
wisconsin5	10.05 ± 3.21	wisconsin10	9.06 ± 3.87
machine5	96.89 ± 0.96	machine10	95.03 ± 0.83
toy	64.45 ± 15.23	auto10	93.72 ± 4.86
auto5	94.78 ± 4.10	housing10	90.31 ± 0.77
housing5	89.98 ± 6.22	stock10	56.49 ± 23.05

5. Conclusions and future work

This paper explores the concept of the empirical feature space (an isomorphic space to the original feature space induced by the kernel trick) to reformulate a well-known ordinal regression method (the Proportional Odds Model or POM) in order to handle nonlinearly separable classification tasks. Different ideas are considered, such as the optimisation of the kernel matrix for tackling ordinal information and the optimisation of the dimensionality of the reduced empirical feature space. These proposals can be used to easily kernelise any existing liner ordinal regression method, independently of their formulation. The different experiments show that the proposed kernel techniques are able to increase the performance of linear ordinal regression methods, such as the POM and reach the performance of the state-of-the-art methods, while still being able to derive natural probability estimates. As future work, several promising lines can be introduced. Firstly, given the connection between our proposal and the Nyström approximation [31], we plan to reformulate this methodology in order to deal with large-scale datasets (by considering the steps followed for the Nyström method approximation). Note that our method, as it is at the moment, may be unaffordable for some large-scale problems, given the use of the singular value decomposition over the complete Gram matrix. Furthermore, because of the good synergy between the kernel learning technique and the proposed ordinal weight matrix, other ordinal kernel algorithms can also be used to analyse its performance.

Acknowledgments

This work has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain). Manuel Cruz-Ramírez's research has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science), grant reference AP2009-0487.

References

[1] A. Agresti, Categorical Data Analysis, 2nd Edition, Wiley Series in Probability and Statistics, Wiley-Interscience, 2002.

[2] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, C. Hervás-Martínez, An Experimental Study of Different Ordinal Regression Methods and Measures, in: 7th International Conference on Hybrid Artificial Intelligence Systems (HAIS), 2012, pp. 296–307.

[3] P. McCullagh, Regression models for ordinal data, Journal of the Royal Statistical Society 42 (2) (1980) 109–142.

[4] W. Chu, S. S. Keerthi, Support vector ordinal regression, Neural Computation 19 (2007) 792–815.

[5] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, W.-B. Li, Kernel discriminant learning for ordinal regression, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 906–910.

[6] J. S. Cardoso, J. F. P. da Costa, Learning to classify ordinal data: The data replication method, Journal of Machine Learning Research 8 (2007) 1393–1429.

[7] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, X. Wang, Ordinal extreme learning machine, Neurocomputing 74 (1–3) (2010) 447–456.

[8] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, A. J. Smola, Input space versus feature space in kernel-based methods, IEEE Transactions on Neural Networks 10 (1999) 1000–1017.

[9] H. Xiong, M. N. S. Swamy, M. O. Ahmad, Optimizing the kernel in the empirical feature space, IEEE Transactions on Neural Networks 16 (2) (2005) 460–474.

[10] S. Abe, K. Onishi, Sparse least squares support vector regressors trained in the reduced empirical feature space, in: Proc. of the 17th international conference on Artificial neural networks, ICANN, Springer-Verlag, 2007, pp. 527–536.

[11] H. Xiong, A unified framework for kernelization: The empirical kernel feature space, in: Chinese Conference on Pattern Recognition (CCPR), 2009, pp. 1 –5.

[12] C. Cortes, M. Mohri, A. Rostamizadeh, Algorithms for learning kernels based on centered alignment, Journal of Machine Learning Research 13 (2012) 795–828.

[13] N. Cristianini, J. Kandola, A. Elisseeff, J. Shawe-Taylor, On kernel-target alignment, in: Advances in Neural Information Processing Systems 14, MIT Press, 2002, pp. 367–373.

[14] J. Verwaeren, W. Waegeman, B. De Baets, Learning partial ordinal class memberships with kernel-based proportional odds models, Comput. Stat. Data Anal. 56 (4) (2012) 928–942.

[15] G. S. Kimeldorf, G. Wahba, Some results on Tchebycheffian spline functions, Journal of Mathematical Analysis and Applications 33 (1) (1971) 82–95.

[16] M. J. Mathieson, Ordinal models for neural networks, in: J. M. A.-P. N. Refenes, Y. Abu-Mostafa, A. Weigend (Eds.), Proceedings of the Third International Conference on Neural Networks in the Capital Markets, Neural Networks in Financial Engineering, World Scientific, 1996, pp. 523–536.

[17] M. Ramona, G. Richard, B. David, Multiclass feature selection with kernel gram-matrix-based criteria, IEEE Trans. Neural Netw. Learning Syst. 23 (10) (2012) 1611–1623.

[18] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 460–474.

[19] J. Zhu, T. Hastie, Kernel logistic regression and the import vector machine, Journal of Computational and Graphical Statistics 14 (1) (2001) 185–205.

[20] Y. Guermeur, A. Lifchitz, R. Vert, Kernel for protein secondary structure prediction, The MIT Press, Cambridge, Massachussetts, 2004, p. 416.

[21] S. Baccianella, A. Esuli, F. Sebastiani, Evaluation measures for ordinal regression, in: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 09), Pisa, Italy.

[22] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, P. Gutiérrez, Metrics to guide a multi-objective evolutionary algorithm for ordinal classification, Neurocomputing 135 (2014) 21–31, advances in Learning Schemes for Function Approximation, Selected

papers from the 11th International Conference on Intelligent Systems Design and Applications (ISDA 2011).

[23] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[24] C. Igel, M. Hüsken, Empirical evaluation of the improved rprop learning algorithms., Neurocomputing 50 (2003) 105–123.

[25] T. Glasmachers, Gradient based optimization of support vector machines, Ph.D. thesis (2008).

[26] A. Asuncion, D. Newman, UCI machine learning repository (2007).
URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[27] PASCAL, Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository (2011).
URL `http://mldata.org/`

[28] W. Chu, Z. Ghahramani, Gaussian processes for ordinal regression, Journal of Machine Learning Research 6 (2005) 1019–1041.

[29] W. Chu, S. S. Keerthi, Support vector ordinal regression, Neural Computation 19 (3) (2007) 792–815.

[30] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[31] P. Drineas, M. W. Mahoney, On the Nyström method for approximating a gram matrix for improved kernel-based learning, Journal of Machine Learning Research 6 (2005) 2153–2175.

4.3. Incorporating privileged information to improve manifold ordinal regression

This section contributes a new method for a special kind of supervised learning, which is called learning using privileged information. Vapnik and Vashist recently proposed a framework to apply support vector machines (SVM) to those cases where privileged information is only available during the training phase [110]. This type of information can be found in many learning problems, where training samples present some special features which are not available during test because of their cost or simply because it is not possible. For example, suppose our goal is to find a rule that can predict the outcome y of a treatment in a year given the current symptoms $\mathbf{x}$ of a patient. At the training stage, a doctor can also provide additional information $\mathbf{x}^*$ about the development of symptoms in three months, six months, and nine months [110]. The algorithm in [110] was based on considering a slack model for this privileged information. Given that slacks are only considered during SVM optimisation and not included in the final model, their approach was able to benefit from this privileged information, mainly improving the convergence of the learning algorithm.

We consider a manifold learning approach for ordinal regression. Ordinal manifold learning has been considered in [77, 79] based on the idea of preserving the intrinsic geometry of the data by defining a neighbourhood graph which respects the ordinal nature of the dataset. This graph is used to construct an adjacency matrix by using a generalised radial basis function. The Laplacian matrix is then derived and used for the learning process. A related method is proposed in [78], where several projections are iteratively computed.

In the following paper, we extend the previous ordinal regression manifold approaches [79, 77] by considering privileged information during the neighbourhood graph construction. Under the assumption that privileged features are useful for the classification, this approach would modify the neighbourhood structure to better represent the learning task. Moreover, we also consider a different approach for constructing the final distance matrix (by making use of the Dijkstra algorithm) and include this information into the kernel matrix, in order to apply support vector ordinal regression [26], as opposed to the ordinal discriminant-based projection method in the original proposal. Therefore, two main objectives can be found in this paper: 1) to analyse whether it is feasible to reformulate the notion of similarity for kernel functions when considering an ordinal manifold of the data; and 2) to study if the inclusion of privileged information helps to improve the constructed model. The results of the experiments confirm that privileged information is able to improve generalisation results for almost all the cases considered. The use of manifold distances for the construction of kernel matrices also produces promising results.

Incorporating Privileged Information to Improve Manifold Ordinal Regression

M. Pérez-Ortiz, P. A. Gutiérrez and C. Hervás-Martínez
University of Córdoba, Dept. of Computer Science and Numerical Analysis,
Rabanales Campus, Albert Einstein Building, 14071 Córdoba, Spain
{*i82perom, pagutierrez, chervas*}*@uco.es*

Keywords: Manifold Learning, Ordinal Regression, Privileged Information, Kernel Learning.

Abstract: Manifold learning covers those learning algorithms where high-dimensional data is assumed to lie on a low-dimensional manifold (usually nonlinear). Specific classification algorithms are able to preserve this manifold structure. On the other hand, ordinal regression covers those learning problems where the objective is to classify patterns into labels from a set of ordered categories. There have been very few works combining both ordinal regression and manifold learning. Additionally, privileged information refers to some special features which are available during classifier training, but not in the test phase. This paper contributes a new algorithm for combining ordinal regression and manifold learning, based on the idea of constructing a neighbourhood graph and obtaining the shortest path between all pairs of patterns. Moreover, we propose to exploit privileged information during graph construction, in order to obtain a better representation of the underlying manifold. The approach is tested with one synthetic experiment and 5 real ordinal datasets, showing a promising potential.

1 INTRODUCTION

Ordinal regression is a learning task where the objective is to classify patterns into a set of predefined labels, but the labels include an order (Cardoso and da Costa, 2007; Chu and Keerthi, 2007; Li and Lin, 2007). For example, for an age estimation problem, people images could be classified into the classes $\{newborn, baby, young, adult, senior\}$. These categories are reflecting intervals of an actual latent variable (the real age of the person) but, contrary to standard regression, the latent variable is unobservable. On the other hand, the order between the categories makes this problem different from standard classification, and specific ordinal regression algorithms try to improve the quality of the classifier by introducing the order in the model and/or penalising the different classification errors (the magnitude of the error should be higher when the predicted class if further to the actual class) (Lin and Li, 2012).

Different methods have been proposed to deal with ordinal regression problems. Threshold models are one of the most popular approaches (McCullagh, 1980; Verwaeren et al., 2012), where the ordinal regression problem is formulated as the problem of estimating a real valued function and a set of $Q-1$ thresholds (Q is the number of classes), in such a way that one interval is assigned to each class ($[-\infty, b_1), [b_1, b_2), \ldots, [b_{Q-1}, \infty)$). This is the structure of the first specific model for ordinal regression, the proportional odds model (McCullagh, 1980), which is an ordinal version of binary logistic regression. Later on, nonlinear threshold models have appeared in the machine learning community, including different adaptions of other methods to the ordinal setting, such as support vector machines (R. Herbrich and Obermayer, 2000; Shashua and Levin, 2003; Chu and Keerthi, 2007), discriminant analysis (Sun et al., 2010) or Gaussian processes (Chu and Ghahramani, 2005). Other works decompose the original ordinal regression problem into several binary classification ones, by sequentially dividing the ordinal scale in binary labels (Frank and Hall, 2001; Cheng et al., 2008; Deng et al., 2010). Finally, a reduction framework can be found in (Cardoso and da Costa, 2007; Lin and Li, 2012), where ordinal regression is reduced to binary classification, but learning one single model for the binary problem where the input patterns are replicated, extended and weighted according to the ordinal label.

In this paper, we consider a manifold learning approach for ordinal regression. The idea of manifold

learning is to uncover the nonlinear structure embedded in a dataset, assuming that the high-dimensional observations lie on or close to an intrinsically low-dimensional manifold. There are different algorithms to learn this kind of structures, including the isometric feature mapping (Isomap) (Tenenbaum et al., 2000) or Laplacian eigenmaps (Belkin and Niyogi, 2001). Based on them, other manifold learning algorithms have been also proposed for classification, such as locality preserving projections (He and Niyogi, 2003) or the discriminant Laplacian embedding (DLE) (Wang et al., 2010).

In the context of ordinal regression, manifold learning has been considered in (Liu et al., 2011a; Liu et al., 2011b) based on the idea of preserving the intrinsic geometry of the data via the definition of a neighbourhood graph which also preserves the ordinal nature of the dataset. This graph is used to construct an adjacency matrix by using a generalised radial basis function. The Laplacian matrix is then derived and used for the learning process. A related method is proposed in (Liu et al., 2012), where several projections are iteratively computed. Finally, ranking on data manifolds is investigated in (Zhou et al., 2004), although the problem is defined as ranking, which is different from ordinal regression.

On the other hand, Vapnik and Vashist recently proposed a framework to apply support vector machines (SVM) to those cases where privileged information is available during the training phase, but not during test (Vapnik and Vashist, 2009). This kind of information can be found in many learning problems, where training samples present some special features which are not available during test because of their cost or simply because it is not possible. For example, suppose our goal is to find a rule that can predict outcome y of a treatment in a year given the current symptoms $\mathbf{x}$ of a patient. At the training stage, a doctor can also provide additional information $\mathbf{x}^*$ about the development of symptoms in three months, six months, and nine months (Vapnik and Vashist, 2009). The algorithm in (Vapnik and Vashist, 2009) was based on considering a slack model for this privileged information. Given that slacks are only considered during SVM optimisation and not included in the final model, their approach was able to benefit from this privileged information, mainly improving the convergence of the learning algorithm.

In this paper, we extend the ordinal regression manifold approach in (Liu et al., 2011b; Liu et al., 2011a) by considering privileged information during the neighbourhood graph construction. Under the assumption that privileged features are useful for the classification task, this approach would modify the neighbourhood structure to better represent the learning task. Moreover, we also consider a different approach for constructing the final distance matrix (by making use of the Dijkstra algorithm) and include this information into a kernel function, in order to apply support vector ordinal regression (Chu and Keerthi, 2007), as opposed to the ordinal discriminant-based projection method in the original proposal. Therefore, two main objectives can be found in this paper: Firstly, to analyse whether it is feasible to reformulate the notion of similarity for kernel functions when considering an ordinal manifold of the data and secondly, to study if the inclusion of privileged information helps to improve the constructed model. The approach is tested in one synthetic dataset and 5 real ones, showing a competitive performance.

The rest of the paper is organised as follows: Section 2 presents the methodology proposed, while Section 3 presents and discusses the experimental results. The last section summarises the main contributions of the paper.

2 METHODOLOGY

When dealing with multiclass classification, the goal is to assign an input vector $\mathbf{x}$ to one of Q discrete classes $\mathcal{C}_q, q \in \{1,\dots,Q\}$. To obtain the prediction rule $C : \mathcal{X} \to \mathcal{Y}$, we use an i.i.d. training sample $X = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ where N is the number of training patterns, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, $\mathcal{X} \subset \mathbb{R}^d$ is the d-dimensional input space and $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ is the label space. We are also provided a test set to obtain a reliable estimation of the classification error, $X_{\mathrm{t}} = \{\mathbf{x}_{\mathrm{t}i}, y_{\mathrm{t}i}\}_{i=1}^{N_{\mathrm{t}}}$, where N_{t} is the number of test patterns and $\mathbf{x}_{\mathrm{t}i} \in \mathcal{X}$, $y_{\mathrm{t}i} \in \mathcal{Y}$. Finally, many learning problems present some features which are available during training but not in the test phase. This privileged information complements training data in such a way that the training sample is $X = \{\mathbf{x}_i, \mathbf{x}_i^*, y_i\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{x}_i^* \in \mathcal{X}^*$, $y_i \in \mathcal{Y}$ and $\mathcal{X}^* \subset \mathbb{R}^{d^*}$ is the d^*-dimensional input privileged space. The test set is the same, given that privileged information is not available when applying the classifier.

Ordinal regression or ordinal classification are those problems where patterns have to be classified into naturally ordered labels. Consequently, the definition of this kind of problems is similar to the one introduced in the previous paragraph, but incorporating the following constraint: $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$, where $\prec$ denotes this order information.

Considering this ordering scale, one of the main hypothesis in ordinal regression is that the distance to adjacent classes is lower than the distance to non-

adjacent classes. Therefore, it can be said that ideally there exists a latent distance-based manifold of the output variable that results in $\mathcal{C}_q$ lying in the space between $\mathcal{C}_{q-1}$ and $\mathcal{C}_{q+1}$. In this paper, we test two different hypotheses. On the one hand and motivated by the large amount of ordinal kernel methods in the literature (Chu and Ghahramani, 2005; Chu and Keerthi, 2007; Sun et al., 2010; Liu et al., 2012), we test whether it is possible to include the manifold structure in the kernel matrix of kernel methods. Kernel matrices can be seen as structures of data that contain information about similarities among the patterns in a dataset. This notion of similarity is usually based on a distance relation between the patterns. Therefore, this distance can be modified to consider the manifold structure of the data. On the other hand, we test whether the inclusion of privileged information in the construction of the neighbourhood graph helps to improve the robustness and efficiency of the classification model. The following two subsections are related to the first hypothesis, while the last subsection covers the second one.

2.1 Constructing a Representative Graph for the Ordinal Manifold

This subsection comprises some elementary notions for constructing a representative graph for the ordinal manifold, which are used both in this paper and the previous work (Liu et al., 2011a; Liu et al., 2011b). Consider an undirected graph of N vertices, $G = (V,E)$, where V corresponds to the vertices of the graph and $E \subseteq [V]^2$ to the edges. In this case, the set of the training patterns form the set of vertices, $V = \{v_1, v_2, \ldots, v_N\} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, and the different edges connect pairs of patterns:

$$E = \{e_{i,j}\} = \{(v_i, v_j)\} = \{(\mathbf{x}_i, \mathbf{x}_j)\}, \qquad (1)$$

where $1 \leq i \leq N$ and $1 \leq j \leq N$. The set of edges is obtained via a k-neighbourhood analysis of the data, i.e. v_i is connected to v_j if $\mathbf{x}_i$ is one of the k-nearest neighbours of $\mathbf{x}_j$ *or* viceversa. Instead of this *or*, we could have considered the logical operator *and*, but we introduce this relaxed version of the neighbouring structure to prevent unconnected regions in the dataset. Note that if v_i is connected to v_j, there exist $e_{i,j}$ such that $e_{i,j} \in E$. For the purpose of constructing the neighbourhood graph, the Euclidean distance is used as the weight function (i.e. the one used for the neighbourhood analysis):

$$f(e_{i,j}) = d(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2, \qquad (2)$$

being $||\cdot||_2$ the L_2-norm operator.

As we aim to preserve the ordinal structure of the manifold, we could try to enlarge the locality between different ranks, as done in (Liu et al., 2011b). To do so, we can include a weight parameter w for the distances in such a way that these weights reflect the rank differences between data points:

$$w_{i,j} = |y_i - y_j| + 1. \qquad (3)$$

This weight information is applied to the distance function as follows: $d(\mathbf{x}_i, \mathbf{x}_j) = w_{i,j} \cdot ||\mathbf{x}_i - \mathbf{x}_j||_2$. The possibility of considering these weights is explored in the experiments of this paper (i.e. we consider both the weighted and unweighted versions of the proposal). Recall that this transformation of the distances is done before constructing the neighbourhood graph.

2.2 Including Graph Shortest Paths in the Kernel Matrix

Usually, for manifold learning algorithms, an adjacency matrix is used for the learning process (which is the underlying idea in (Liu et al., 2011a; Liu et al., 2011b)). In this paper, however, we try to analyse whether it is feasible to reformulate the notion of similarity for kernel functions when considering an ordinal manifold of the data. The main idea is to use the graph information obtained in the previous step to locate the different patterns in the underlying ordinal manifold of the data. To do so, we use the shortest path of the graph in order to provide a more smooth approach for the distances (as opposed to other manifold-based techniques where non-connected points are assumed to present an infinite distance).

In graph theory, the shortest path problem is the problem of finding a path between two vertices in a graph such that the sum of the weights of its constituent edges is minimised. As said, the constructed graph is undirected, so the notion of path is defined as a sequence of z vertices from v_1 to v_z, $p_{1z} = (v_1, v_2, \ldots, v_z) \in V^z$, such that v_i is adjacent to v_{i+1} for $1 \leq i < z$ (and therefore $e_{i,i+1}$ exists). Moreover, given a real-valued weight function $f : E \to \mathbb{R}$ (as said, the weighted or unweighted Euclidean distance) that assigns a cost to each edge and an undirected graph G, the shortest path from v to v' is the path $p_{1,z} = (v_1, \ldots, v_z)$ (where $v_1 = v$ and $v_z = v'$) that over all possible paths minimises the sum $\sum_{i=1}^{z-1} f(e_{i,i+1})$, where $e_{i,i+1} \in E$.

To compute the distance from one data pattern $\mathbf{x}_i$ to the rest but taking into account the manifold structure, we can compute the shortest paths from the vertex v_i to all the rest of vertices considering the well-known Dijkstra's algorithm (Dijkstra, 1959). Denote by P the set of paths obtained from this process, where

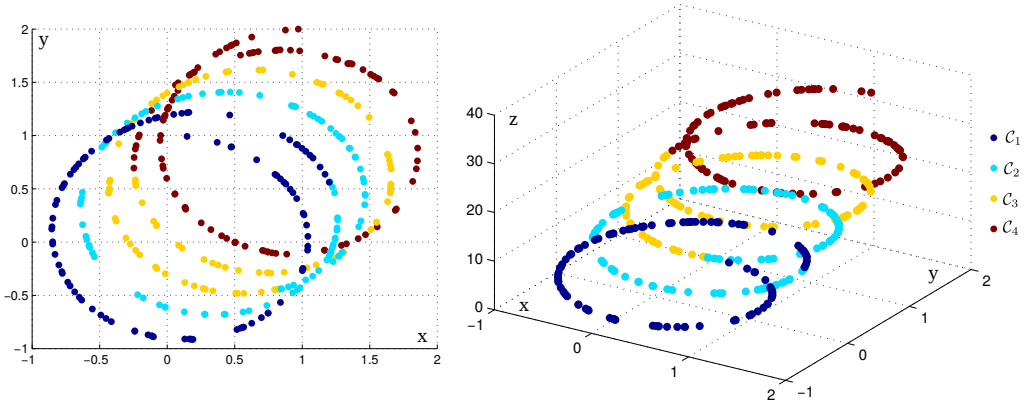


Figure 1: Representation of the `spiral` synthetic ordinal dataset. Left plot: Original dataset without privileged information. Right plot: Dataset including the privileged information as an additional feature. It can be seen that this privileged information improves the potential separability of the data.

$p_{i,j} \in P$ is the shortest path between v_i and v_j. Therefore, the distance from any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the training set is:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{h=1}^{z-1} f(e_{h,h+1}), v_1 = \mathbf{x}_i, v_z = \mathbf{x}_j. \quad (4)$$

where z is the length of the path between $\mathbf{x}_i$ and $\mathbf{x}_j$. Note that $d(\mathbf{x}_i, \mathbf{x}_j) = w_{i,j} \cdot ||\mathbf{x}_i - \mathbf{x}_j||_2$ if $\mathbf{x}_i$ is one of the nearest neighbours of $\mathbf{x}_j$. Therefore, to introduce the information of the location of each data point in the manifold in the kernel matrix, we modify the kernel function as follows:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right), \quad (5)$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is defined as in Eq. (4) and σ is the kernel parameter, as opposed to using the standard Gaussian kernel with the L_2-norm: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2\sigma^2}\right)$. Note that this kernel matrix will still be positive semidefinite given that the only information changed is the distance function.

The kernel matrix obtained by this process is the one used for the training step. For the test phase, we first compute the distance from each test pattern $\mathbf{x}_{ti}$ to its nearest neighbour in training $\mathbf{x}_j$ and then, sum this distance to the shortest paths from $\mathbf{x}_j$ to the rest of training patterns. Consequently, the distance between a test point $\mathbf{x}_{ti}$ and all training points is:

$$d(\mathbf{x}_{ti}, \mathbf{x}_z) = d(\mathbf{x}_{ti}, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_z), \quad (6)$$

for $1 \leq i \leq N_t$ and $1 \leq z \leq N$.

This idea for the test phase corresponds to locate the test pattern in the graph and use the shortest paths information to compute the distance to the whole set of training patterns.

2.3 Including Privileged Information in the Graph

In order to motivate the inclusion of privileged information during manifold learning, Figure 1 represents a synthetic dataset presenting an ordinal manifold-based structure where the label of points is assigned according to the z coordinate. Data points lie on a leaning 3-dimensional spiral and labels are ordinal, with four classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ and $\mathcal{C}_4$. The Figure 1 also includes the projection over x and y coordinates. As can be seen, z coordinate is crucial to obtain a neighbourhood graph able to help in the ordinal classification task. Considering this z value as privileged information during graph construction would allow the classification of patterns, even when only x and y features are available during the test phase.

The privileged information can be easily included during distance calculation to construct a neighbourhood graph which takes into account this additional information. We can make use of the privileged features in the real-valued weight function f that assigns a value to edges of the graph:

$$\begin{aligned} f^*(e_{i,j}) &= ||(\mathbf{x}_i, \mathbf{x}_i^*) - (\mathbf{x}_j, \mathbf{x}_j^*)||_2 \quad (7) \\ &= \sqrt{\sum_{s=1}^{d} (x_{is} - x_{js})^2 + \sum_{s=1}^{d^*} (x_{is}^* - x_{js}^*)^2}. \end{aligned}$$

The whole process of neighbourhood analysis and shortest path computation is reformulated to work with this real-valued weight function. When considering this weight function, $f^*(e_{i,j})$, the distance function on Eq. (4) will be $d^*\left((\mathbf{x}_i, \mathbf{x}_i^*), (\mathbf{x}_j, \mathbf{x}_j^*)\right)$ and will be applied to the kernel function on Eq. (6). For the test phase, the privileged information is only consid-

ered for the graph that has been previously learnt, i.e.:

$$d\left(\mathbf{x}_{ti},(\mathbf{x}_z,\mathbf{x}_z^*)\right)=||\mathbf{x}_{ti}-\mathbf{x}_j||_2+d^*\left((\mathbf{x}_j,\mathbf{x}_j^*),(\mathbf{x}_z,\mathbf{x}_z^*)\right),$$

where $1\leq z\leq N$ and $\mathbf{x}_j$ is the closest training point from the test point evaluated $\mathbf{x}_{ti}$.

3 EXPERIMENTS

The proposed methodologies are based on generating a modified version of the kernel matrix (by exploiting the neighbourhood graph of the data), so they can be applied to any kernel classifier. In this way, we have considered the Support Vector Ordinal Regression with Implicit Constraints (SVORIM) (Chu and Keerthi, 2007), as it is one of the best performing threshold models for ordinal regression (Gutiérrez et al., 2012). 5 benchmark ordinal regression datasets have been used for the analysis, which are taken from publicly available repositories[1] (Asuncion and Newman, 2007; PASCAL, 2011). Additionally, a more controlled environment is provided by the `spiral` dataset, introduced in Section 2.3. Table 1 shows the characteristics of the evaluated datasets, where it can be checked that number of classes varies between 3 and 5.

In the experiments, we evaluate two different factors:

- The introduction of ordinal costs for penalising distances during the construction of the graph. Ordinal costs are based on the absolute cost. This factor will be used to confirm whether these costs are really useful for ordinal regression, as discussed in previous works (Liu et al., 2011b; Liu et al., 2011a).
- The improvement obtained by the privileged information. The graph will be constructed with and without privileged information to evaluate if the additional variables improve the quality of the model.

The most common evaluation measures for ordinal regression are the Mean absolute error (*MAE*) and the accuracy ratio (*Acc*) (Gutiérrez et al., 2012; Baccianella et al., 2009; Cruz-Ramírez et al., 2014). The *MAE* measure is used when the costs of different misclassification errors is not constant:

$$MAE=\frac{1}{N_{\mathrm{t}}}\sum_{i=1}^{N_{\mathrm{t}}}|y_{ti}-\hat{y}_{ti}|, \tag{8}$$

[1]Note that many of these datasets are frequently treated as nominal ones, without taking into account the order scale.

where $\hat{y}_{ti}$ is the label predicted for $\mathbf{x}_{ti}$. *MAE* values range from 0 to $Q-1$ (Baccianella et al., 2009).

Regarding the experimental setup, the datasets were divided 30 times using a holdout stratified technique with a 75% of the patterns for training and the remaining 25% for test. The splits of each holdout are the same for all the algorithms and one model is obtained for each training set and evaluated in the test set. The average test evaluation measures and the corresponding standard deviations are finally reported as the summary of the algorithm performance.

We use the standard Gaussian kernel for all the methods. Model selection is accomplished by cross-validating the hyperparameters of the algorithms considering only the training data (with a 5-fold cross-validation). The measure used to select the best parameter combination is *MAE*. The two parameters to be optimised are the kernel width (σ) and the cost parameter (C), both being selected within the values $\sigma, C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. The number of nearest neighbours to be considered during graph construction is $k=3$. In those cases that a pattern is not connected to any other ones for the current value of k, we increase k until all patterns are connected to at least one.

For the `spiral` dataset, privileged information is the z coordinate. For the rest of datasets, we apply the Relief feature selection algorithm (Kira and Rendell, 1992) over the training set to sort the features by their relevance. We select half of the features (the most relevant ones) as privileged information ($\mathbf{x}^*$) and the rest as the original information ($\mathbf{x}$).

3.1 Results

Table 2 shows the test results for the 6 ordinal datasets considered in terms of *Acc* and *MAE*. The best result for each dataset is in bold face and the second one in italics. From this Table, we can outline several conclusions:

- When considering the ordinal weights, the *Acc* and *MAE* results are always improved by the privileged information. However, if the costs are not included, there are some datasets where the privileged information does not improve the results (`bondrate`, `contact-lenses` and `squash-unstored`). Given that the cross-validation criterion is the *MAE* (which is based on an absolute cost loss), we conclude that using these weights is necessary to properly obtain a benefit from the privileged information.
- From all the combinations, considering privileged information and ordinal weights is the best one,

Table 1: Characteristics of the six datasets used for the experiments: number of instances (Size), inputs (#In.), classes (#Out.) and patterns per-class (#PPC)

Dataset	Size	#In.	#Out.	#PPC
bondrate	57	37	5	(6,33,12,5,1)
contact-lenses	24	6	3	(15,5,4)
pasture	36	25	3	(12,12,12)
spiral	400	3	4	(50,50,50,50)
squash-unstored	52	52	3	(24,24,4)
tae	151	54	3	(49,50,52)

Table 2: Test results obtained for the different datasets (Mean± Standard Deviation of the 30 splits) by considering all the different manifold classification algorithms based on SVORIM.

		Acc		*MAE*	
		Privileged Information		Privileged Information	
Dataset	Ordinal Weights	No	Yes	No	Yes
bondrate	No	*57.28 ± 3.82*	56.54 ± 6.50	*0.6272 ± 0.0647*	0.6296 ± 0.0893
	Yes	56.54 ± 5.02	**58.52 ± 5.34**	0.6346 ± 0.0676	**0.6123 ± 0.0996**
contact-lenses	No	*61.11 ± 10.11*	*61.11 ± 10.11*	*0.5500 ± 0.0892*	*0.5500 ± 0.0892*
	Yes	58.89 ± 12.17	**62.22 ± 8.68**	0.5722 ± 0.1132	**0.5389 ± 0.0717**
pasture	No	*48.89 ± 14.76*	**51.85 ± 14.69**	*0.5370 ± 0.1600*	**0.5074 ± 0.1450**
	Yes	42.96 ± 15.91	43.70 ± 16.49	0.6037 ± 0.1668	0.6000 ± 0.1716
spiral	No	82.37 ± 4.16	*87.80 ± 2.70*	0.2260 ± 0.0589	*0.1867 ± 0.0505*
	Yes	85.03 ± 3.62	**87.90 ± 2.76**	0.2120 ± 0.0567	**0.1857 ± 0.0520**
squash-unstored	No	**52.56 ± 13.71**	50.77 ± 10.42	*0.4795 ± 0.1452*	0.4949 ± 0.1082
	Yes	49.74 ± 9.84	*51.54 ± 11.45*	0.5077 ± 0.0960	**0.4897 ± 0.1115**
tae	No	*35.35 ± 8.62*	**35.53 ± 8.40**	0.6570 ± 0.0867	*0.6526 ± 0.0783*
	Yes	34.91 ± 6.66	**35.53 ± 8.40**	0.6754 ± 0.0783	**0.6500 ± 0.0770**

obtaining the best results in four datasets and the second one in another.

- The most clear contribution of the privileged information is obtained for the spiral dataset. This is due to the fact in this more controlled environment data clearly belong to a low dimensional manifold and the class label is assigned according to the privileged information (z value). For the rest of datasets, the privileged information has been selected according to the Relief algorithm, which has known limitations. Nevertheless, there are some datasets where the contribution of privileged information is still quite noticeable (e.g. bondrate and contact-lenses).
- The original SVORIM algorithm (without using a manifold assumption) was run for the spiral dataset and the same configuration, leading to a performance of $Acc = 78.80 \pm 3.53$ and $MAE = 0.2617 \pm 0.0467$. It is noticeable that these values are worse than the ones obtained by the manifold proposals in this paper.

4 CONCLUSIONS

This paper considers a new approach to face ordinal regression problems based on manifold learning. This approach is based on constructing a neighbourhood graph with the purpose of obtaining the intrinsic structure of the data. The main paper contribution is that this neighbourhood graph can be improved by the use of privileged information, information that is available during training but not in the test phase.

The algorithm is applied to 5 ordinal classification real problems and one synthetic dataset. When combined with SVORIM, the results of this paper confirm that privileged information is able to improve generalisation results for almost all the cases considered. The distances used in the kernel matrices are obtained using the privileged features, which (under the assumption that privileged information is really informative) better reflects the data structure.

Several future research directions are still open from the work in this paper. First of all, more datasets should be considered, including datasets with a higher number of patterns and with a more clear manifold

structure. For example, the experiments considered in (Liu et al., 2011b) cover the UMIST face, MovieLens and the USPS datasets, which are known to contain an underlying manifold structure. The problem is that meaningful privileged information has to be found for these problems. Secondly, the methods should be compared against standard manifold classifiers to check their performance. Finally, alternative kernel methods apart from SVORIM could be considered together with the proposals in this paper.

ACKNOWLEDGEMENTS

This work has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain).

REFERENCES

Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

Baccianella, S., Esuli, A., and Sebastiani, F. (2009). Evaluation measures for ordinal regression. In *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA 09)*, pages 283–287, Pisa, Italy.

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591.

Cardoso, J. S. and da Costa, J. F. P. (2007). Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research*, 8:1393–1429.

Cheng, J., Wang, Z., and Pollastri, G. (2008). A neural network approach to ordinal regression. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2008, IEEE World Congress on Computational Intelligence)*, pages 1279–1284. IEEE Press.

Chu, W. and Ghahramani, Z. (2005). Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041.

Chu, W. and Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation*, 19(3):792–815.

Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., and Gutiérrez, P. A. (2014). Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing*, 135:21–31.

Deng, W.-Y., Zheng, Q.-H., Lian, S., Chen, L., and Wang, X. (2010). Ordinal extreme learning machine. *Neurocomputation*, 74(1-3):447–456.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Frank, E. and Hall, M. (2001). A simple approach to ordinal classification. In *Proc. of the 12th Eur. Conf. on Machine Learning*, pages 145–156.

Gutiérrez, P. A., Pérez-Ortiz, M., Fernandez-Navarro, F., Sánchez-Monedero, J., and Hervás-Martínez, C. (2012). An Experimental Study of Different Ordinal Regression Methods and Measures. In *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, volume 7209 of *Lecture Notes in Computer Science*, pages 296–307.

He, X. and Niyogi, P. (2003). Locality preserving projections. In *NIPS*, volume 16, pages 234–241.

Kira, K. and Rendell, L. A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134.

Li, L. and Lin, H.-T. (2007). Ordinal Regression by Extended Binary Classification. In *Advances in Neural Inform. Processing Syst. 19*.

Lin, H.-T. and Li, L. (2012). Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367.

Liu, Y., Liu, Y., and Chan, K. C. C. (2011a). Ordinal regression via manifold learning. In Burgard, W. and Roth, D., editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 398–403. AAAI Press.

Liu, Y., Liu, Y., Chan, K. C. C., and Zhang, J. (2012). Neighborhood preserving ordinal regression. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service (ICIMCS12)*, pages 119–122, New York, NY, USA. ACM.

Liu, Y., Liu, Y., Zhong, S., and Chan, K. C. (2011b). Semi-supervised manifold ordinal regression for image ranking. In *Proceedings of the 19th ACM international conference on Multimedia (ACM MM2011)*, pages 1393–1396, New York, NY, USA. ACM.

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society*, 42(2):109–142.

PASCAL (2011). Pascal (pattern analysis, statistical modelling and computational learning) machine learning benchmarks repository.

R. Herbrich, T. G. and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press.

Shashua, A. and Levin, A. (2003). Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems (NIPS)*, pages 937–944. MIT Press, Cambridge.

Sun, B.-Y., Li, J., Wu, D. D., Zhang, X.-M., and Li, W.-B. (2010). Kernel discriminant learning for ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 22:906–910.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Vapnik, V. and Vashist, A. (2009). A new learning

paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557.

Verwaeren, J., Waegeman, W., and De Baets, B. (2012). Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational Statistics & Data Analysis*, 56(4):928–942.

Wang, H., Huang, H., and Ding, C. H. (2010). Discriminant laplacian embedding. In *AAAI*.

Zhou, D., Weston, J., Gretton, A., Bousquet, O., and Schölkopf, B. (2004). Ranking on data manifolds. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS2003)*, pages 169–176.

Everything is theoretically impossible, until it is done.

Robert A. Heinlein

5

Over-sampling techniques for the imbalanced nature of ordinal problems

This chapter presents different works related to the topic of over-sampling, which is a method for alleviating the effect of the natural imbalance of some classification problems. The former paper in this chapter includes a study on the topic of over-sampling in the feature space induced by kernel functions. The latter proposes a reformulation of over-sampling methods for ordinal regression problems.

Main publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez, P. Tino and C. Hervás-Martínez. Over-sampling the minority class in the feature space. Submitted to *IEEE Transactions on Neural Networks and Learning Systems* (Under Review), 2014, Impact Factor (2013): 4.370 (Q1).
- M. Pérez-Ortiz, P.A. Gutiérrez, C. Hervás-Martínez and X. Yao. Graph-Based Approaches for Over-sampling in the context of Ordinal Regression. *IEEE Transactions on Knowledge and Data Engineering* (TKDE), In press, 2015, Impact Factor (2013): 1.815 (Q1).

Other publications associated to this chapter:

- M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Synthetic over-sampling in the empirical feature space. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), pages 385–390, 2013.
- M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez. Borderline kernel based over-sampling. In *International Conference on Hybrid Artificial Intelligence Systems* (HAIS), Lecture Notes in Computer Science Volume 8073, pages 472–481, 2013.

The two main publications are now presented in the different subsections of this chapter.

5.1. Over-sampling the minority class in the feature space

As stated in the introduction, most classification methods conveniently assume that the classification problem is balanced, i.e. that the prior class probability distribution is of high entropy, which usually poses serious hindrance for the learning process. To solve this, different methodologies have emerged in the context of machine learning, such as over-sampling and under-sampling (i.e. what is commonly known as data approaches) and cost-sensitive or algorithmic strategies. In this sense, over-sampling techniques have been seen to be one of the most outstanding and successful approaches, despite the fact that their formulation can result in data inconsistencies. The reason for this is that most of them rely on a convex combination of patterns where the classes in general can not be assumed to be convex [23], and therefore they can produce synthetic patterns within the majority class region.

To solve this, the following paper proposes the over-sampling of patterns in the feature space induced by a kernel function, where ideally, if the kernel fits the data, the classes will be linearly separable. Since the feature space is not accessible, because the only information available is the dot product between points, the notion of empirical feature space [95, 116] is used for over-sampling purposes (which is a Euclidean space isomorphic to the original feature space). The proposed method is tested in the context of support vector machines (SVM) where imbalanced datasets pose a serious hindrance for learning, although, by definition, the methodology could be applied to any classifier. A flexible kernel learning technique, that maximises the data class separation, is also used to validate the initial hypothesis and to study the influence of the kernel function in the method. Finally, we derive an unified framework for preferential over-sampling (i.e. a framework which analyses the more suitable patterns to be over-sampled) using the optimal SVM hyperplane solution and kernel learning techniques. A thorough set of experiments over 50 binary imbalanced datasets is conducted to validate the main hypotheses of this work. From the results, several conclusions can be drawn: firstly, over-sampling in the empirical

feature space yields better performance than over-sampling in the input space; secondly, that the control of the dimensionality of the empirical feature space could lead to better results due to the concentration of spectral properties; that the kernel used may influence the solution to a great extent; and finally, that there exist some regions of the dataset which should be preferred for over-sampling.

Over-sampling the minority class in the feature space

Journal:	*IEEE Transactions on Neural Networks and Learning Systems*
Manuscript ID:	TNNLS-2014-P-4007
Manuscript Type:	Paper
Date Submitted by the Author:	07-Nov-2014
Complete List of Authors:	Pérez-Ortiz, María; University of Córdoba, Dpt. of Computer Science and Numerical Analysis Gutiérrez, Pedro Antonio; University of Cordoba, Department of Computer Science and Numerical Analysis; Tino, Peter; University of Birmingham, School of Computer Science; Hervas, Cesar; University of Córdoba, Department of Computer Science and Numerical Analysis
Keywords:	over-sampling, imbalanced classification, kernel methods, empirical feature space, support vector machines

Over-sampling the minority class in the feature space

M. Pérez-Ortiz, P.A. Gutiérrez *Member, IEEE,*, P. Tiňo, and César Hervás-Martínez, *Member, IEEE,*

***Abstract*—The imbalanced nature of some real-world data is one of the current challenges for machine learning researchers. One common approach over-samples the minority class through convex combination of its patterns. We explore the general idea of synthetic over-sampling in the feature space induced by a kernel function (as opposed to input space). If the kernel function matches the underlying problem, the classes will be linearly separable and synthetically generated patterns will lie on the minority class region. Since the feature space is not directly accessible, we use the empirical feature space (a Euclidean space isomorphic to the feature space) for over-sampling purposes. The proposed method is framed in the context of support vector machines where imbalanced datasets can pose a serious hindrance. The idea is investigated in three scenarios: 1) over-sampling in the full and reduced-rank empirical feature spaces; 2) a kernel learning technique maximising the data class separation to study the influence of the feature space structure (implicitly defined by the kernel function); 3) a unified framework for preferential over-sampling that spans some of the previous approaches in the literature. We support our investigation with extensive experiments over 50 imbalanced datasets.**

***Index Terms*—Over-sampling, imbalanced classification, kernel methods, empirical feature space, support vector machines**

I. Introduction

Classification methods in machine learning often conveniently assume that the prior class probability distribution is of high entropy. However, this is not the case in many real-world applications from areas such as medical diagnosis, information retrieval, fraud detection, fault monitoring, etc. The classification paradigm when one or several classes have a much lower prior probability in the training set is known as imbalanced classification [1], [2] and it poses a difficult challenge for machine learning researchers. Because of that, imbalanced classification is currently receiving a lot of attention from the pattern recognition and machine learning communities [3]–[9]. Often, the minority class happens to be more important than the majority one, but it may also be much more difficult to model due to the low number of available samples. Since most traditional learning systems have been designed to work on balanced data, they will usually be focused on improving overall performance and be biased towards the majority class, consequently harming the minority one [10]. Although from a formal definition an imbalanced dataset is any set of labelled data exhibiting an unequal distribution between classes, it has been shown that this is not the only factor involved hindering the learning in this context [1], [2]. The complexity of the data (existence of noisy and non-representative samples or class overlapping) or the size of the training set (high-dimensional data or small sample size) can also be part of the nature of the class imbalance problem. The approaches developed over the years for tackling the class imbalance problem can be categorised in two groups:

- Data approach - based on sampling methods, including over-sampling minority groups (groups of interesting rare examples), or under-sampling majority groups (groups with large example sizes), the combination of both being also very popular [1].
- Algorithm approach - forces the classifier to pay more attention to the minority class, e.g. by a cost-sensitive approach [11].

The analysis made in this paper is contextualised on data approaches. Thus, a brief discussion on these techniques is now given (for a detailed review of over-sampling techniques see [1]). Roughly speaking, it can be said that over-sampling and under-sampling are opposite and equivalent techniques at the same time, since they are aimed at the same purpose (i.e., balance the class distribution) but using different approaches. Formally, over-sampling concerns to the process of sampling a distribution with a significantly higher frequency than the given one and under-sampling to the process of reducing the frequency of the majority class. As said, in both cases, the methodologies impose a balance in the class distribution in order to avoid aliasing and focus on the classification of minority classes. Although both over-sampling and under-sampling approaches have been shown to improve classifier performance over imbalanced datasets, different studies suggest that over-sampling is more useful than under-sampling [2], [12], specially for highly imbalanced and complex datasets. Recall that under-sampling could entail a loss of potentially meaningful information of the dataset.

Concerning over-sampling, the first idea is to perform a random replication of the minority data, but this often leads to over-fitting [10]. Another common approach is to generate new synthetic patterns according to the minority class distribution. One of the most well-known methods in this group is the synthetic minority over-sampling technique (SMOTE) [3] based on generating new instances by convex combination of one point and one of its k-nearest neighbours (both belonging to the minority class). However, the classes in general cannot be assumed to be convex and hence SMOTE does not avoid

The work of M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain). The work of P. Tino has been supported by EPSRC grant EP/L000296/1. M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez are with the Department of Computer Science and Numerical Analysis of the University of Córdoba, Spain, email: {i82perom,pagutierrez,chervas}@uco.es. P. Tiňo is with the School of Computer Science of the University of Birmingham, Birmingham, United Kingdom, email: p.tino@cs.bham.ac.uk

new synthetic patterns to fall inside majority regions, therefore, more careful techniques have been developed to prevent this issue (prevent, but not solve). Adaptive synthetic [5]–[7] and cluster-based sampling methodologies [8], [9] are examples of more powerful techniques, based on extracting knowledge directly from the data to analyse which patterns and regions of the space are more suitable for over-sampling. This will be referred in the paper to as preferential over-sampling. At the same time, kernel methods have been spreading rapidly and gaining more acceptance from machine learning researchers due to their good generalisation ability and their determinism, being one of the most widely used the Support Vector Machine paradigm (SVM) [13], [14]. However, for the specific SVM technique, imbalanced data pose a serious challenge, due to the formulation of the soft-margin maximisation paradigm which is focused on improving overall performance. Thus, the combination of kernel methods with techniques for tackling class imbalance is widely spread [4], [15].

It is clear that over-sampling by linear interpolation is not as suitable when dealing with nonlinear classifiers as it could be than when applying linear classifiers. However, linearly separable datasets are not common in real-world applications, thus making advisable the application of classifiers able to capture this nonlinearity. Besides, the development of a suitable nonlinear over-sampling strategy could be tricky. Thus, in contrast to previous approaches, we propose to generate new synthetic data by convex combination of points in a space where the classes are (ideally) linearly separated - making generation of new synthetic points by convex combination of the original points belonging to the same class safe. This is done using the feature space induced by a kernel function for over-sampling the patterns rather than using the input space. However, this is not so straightforward, because when dealing with kernel methods the only information available is the dot products of the images of the patterns [16]. To cope with this issue, this paper makes use of the notion of the empirical feature space (EFS) [17], [18], which is Euclidean and preserves the geometrical structure of the original feature space, given that distances and angles in the feature space are uniquely determined by dot products and that the dot products of the corresponding images are the original kernel values.

Thus, the main motivation for performing over-sampling in the EFS (instead of in the input space) is the hypothesis that the feature space will provide a more suitable space for over-sampling (at least, for convex combination of points) because the separation of the classes will be simpler and larger (ideally, due to the kernel trick they will be linearly separable). At the same time, this technique can be seen as a general nonlinear over-sampling in the input space due to the application of the nonlinear map Φ related to the kernel trick and could be used in combination with any classifier.

To the best of our knowledge, the idea of performing over-sampling in the feature space has only been researched before in [15] (recall that in our case, it is performed in the EFS). In this previous work, the synthetic instances were generated by using the geometric interpretation of the dot products in the kernel matrix, and the pre-images of the synthetic instances were approximated based on a distance relation between the feature space and the input one, since inverse mapping $\Phi(\cdot)^{-1}$ from the feature space to the input space is not available. Our proposal is free of the assumptions of this inverse mapping approximation.

The study made in this paper is intended to provide an extensive analysis of over-sampling in the EFS and can be subdivided in three sections. The first one deals with the issue of extending the SMOTE over-sampling algorithm to be used in the full and reduced-rank EFS. The objective is to test whether the EFS provides a more suitable framework for over-sampling by convex combination of patterns and to deal with the dimensionality of the EFS. The second part deals with the kernel function choice (since our over-sampling methodology will obviously depend on how the kernel matches the underlying classification problem) and we develop a flexible strategy for optimising the feature space structure based on analytical knowledge (using the notion of kernel-target alignment [19], [20]). Ideally, a better fitted kernel function will increase the separability of the classes, providing us with a '*safer*' environment for the generation of synthetic patterns by convex combination. The last part of this paper proposes a unified adaptive framework for preferential over-sampling generalising several over-sampling approaches in the literature [3], [5], [6], making use of the optimal hyperplane of the SVM solution and kernel learning techniques for optimising the synthetically generated patterns. The objective is to check if some regions of the space can be more useful for over-sampling than others. To test the different hypotheses exposed in this paper, we perform a thorough set of experiments with 50 binary and imbalanced datasets.

The paper is organized as follows: Section II introduces some useful notions; Section III exposes how to perform over-sampling in the EFS; Section IV develops a new methodology for kernel learning; Section V proposes a general preferential over-sampling framework; Section VI exposes the experimental study and analyses the results obtained; and finally, Section VII outlines some conclusions and future work.

II. Background

This section is intended to introduce the notation used throughout all the paper and to provide some previous notions about SVM classifiers and the empirical feature space.

Consider a sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^m \subseteq \mathcal{X} \times \mathcal{Y}$ generated i.i.d. from a (unknown) joint distribution $P(\mathbf{x}, y)$, where $\mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} = \{+1, -1\}$. The goal in binary classification is to assign an input vector $\mathbf{x}$ to one of 2 classes $\{+1, -1\}$. Denote by $X^{\mathtt{tr}}$ and $X^{\mathtt{ts}}$ the sets of training and testing inputs, respectively. Furthermore, we will mark by subscript $+$ and $-$ to the sets containing inputs from the positive and negative class, respectively. For a set X, we denote by $\mathbf{X}$ the design matrix storing points of X as rows.

Reproducing kernels (often referred as Mercer kernels) are functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which for all pattern sets $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ give rise to semidefinite positive matrices $\mathbf{K}_{m \times m}$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Kernel functions allow us to derive nonlinear classifiers by reducing them to linear ones but in some Hilbert space $\mathcal{H}$ nonlinearly related to the

input space and furnished with a dot product $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$. The use of this kernel function instead of the dot product in $\mathbb{R}^m$ corresponds to using a (usually) nonlinear mapping of patterns from $\mathcal{X}$ to a high-dimensional or infinite-dimensional Hilbert Space $\mathcal{H}$ such that $\Phi : \mathcal{X} \to \mathcal{H}$, where the separation would ideally be easier, and take the dot product there. Kernel machines trained on D do not operate on the whole of $\mathcal{H}$ but on its subset $\mathcal{F} = \text{span}\{\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_m)\}$, which we will refer to as the feature space such that $\mathcal{F} \subset \mathcal{H}$. Note that $\mathcal{F}$ is at most an m-dimensional linear space.

A. Support Vector Machines

SVM [13], [14] is perhaps the most common kernel learning method for statistical pattern recognition due to its good generalisation ability and freedom from local minima. The basic idea behind this technique is the separation of two different classes through a hyperplane which is specified by a normal vector $\mathbf{w}$ and a bias b. The optimal separating hyperplane is the one which maximises the distance between the hyperplane and the nearest points in both classes (called margin). Beyond the application of kernel techniques to allow non-linear decision discriminants (the kernel trick), another generalisation was made to replace hard margins with soft margins [14], using the so-called slack-variables ξ_i in order to deal with overlapping classes. Therefore, this algorithm seeks for a classifier $f : \mathbb{R}^d \to \mathbb{R}$ of the form $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ (Φ being the mapping function induced by the kernel) that minimises the objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{m} \xi_i, \tag{1}$$

for some parameter C, subject to the constraints:

$$y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall_i \in \{1, \ldots, m\}.$$

It is clear that, using SVMs, the soft-margin maximization paradigm poses a serious hindrance for imbalanced datasets [21], [22]. The main reason for this is that soft-margin SVM optimisation is focused on overall error, therefore, they are inherently biased toward the majority class. In the worst case, for a noisy and highly imbalanced dataset, the SVM paradigm is very likely to obtain a trivial classifier (i.e., the one that classifies all the patterns in the majority class), a solution that, as said, if the imbalance is severe, could provide the minimal error [1]. To cope with this issue, several studies in the machine learning literature have explored different solutions to the imbalanced classification problem considering the SVM paradigm. Most of them are based on over-sampling [21], under-sampling [4], cost-sensitive classification [23], ensembles [24], [25] and kernel optimisation techniques [26], [27], among others [28], [29]. However, some studies suggest that under-sampling is not as effective as over-sampling in this case because of the potential loss of information on the class boundaries [21], which is crucial for the SVM solution.

B. Synthetic minority over-sampling technique (SMOTE)

As stated before, one of the most widely used techniques for over-sampling is the SMOTE algorithm [3]. The process is very simple: the method consists on generating new instances on the line that connects one randomly chosen point with one of its k-nearest neighbours [30], both belonging to the minority class. Therefore, this methodology relies on a convex combination of two patterns for the generation of the synthetic ones. Note that with this approach new patterns could lie inside the majority class region (although choosing a correct value for the k parameter of the k-nearest neighbours method could avoid this to happen in some cases).

C. Empirical feature space (EFS)

We can endow an r-dimensional ($r \leq m$) space $\mathcal{F}$ with an orthonormal basis $\{\mathbf{u}_g\}, g \in B, B = \{1, 2, \ldots, r\}$, satisfying orthogonality, normalisation and completeness. Consider the set:

$$\mathcal{E} = \{\varphi(\mathbf{v}) | \mathbf{v} \in \mathcal{F}\},$$

where $\varphi(\mathbf{v}) = \{\langle \mathbf{v}, \mathbf{u}_g \rangle_{\mathcal{F}}\}_{g \in B}$. The map φ is an isometric isomorphism of $\mathcal{F}$ and $\mathcal{E}$ [31], i.e. it is a bijective linear mapping such that the dot products are preserved: $\langle \varphi(\mathbf{v}), \varphi(\mathbf{v}') \rangle_{\mathcal{E}} = \langle \mathbf{v}, \mathbf{v}' \rangle_{\mathcal{F}}$. When $\mathcal{F}$ is the feature space, the set $\mathcal{E}$ is referred to as the empirical feature space (EFS).

Consider a set of training points $\{\mathbf{x}_i\}_{i=1}^m \subseteq \mathcal{X}$. Then, when working with kernel methods we use a kernel function k to map the patterns to the feature space $\mathcal{F}$ and thus obtain a Gram matrix $\mathbf{K}$ with rank r, $r \leq m$. The nonlinear map from the input space to the r-dimensional Euclidean space $\Phi_r^e : \mathcal{X} \to \mathbb{R}^r$ which preserves the feature space structure is referred to as the empirical kernel map [17]. The EFS $\mathcal{E}$ is chosen so as to preserve the dot product information about $\mathcal{F}$ contained in $\mathbf{K}$, i.e., to be isometric isomorphic to the embedded feature space $\mathcal{F} \subset \mathcal{H}$. In this sense, it can be said that the empirical kernel map corresponds to a bijective linear mapping $\varphi : \mathcal{F} \to \mathcal{E}$.

A graphical representation of the input space, feature space, EFS and mappings between these spaces is shown in Fig. 1.

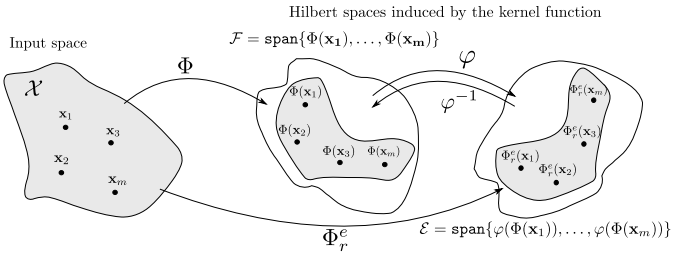


Fig. 1: Representation of the relation and mapping between input space, feature space and empirical feature space.

Any given Gram matrix $\mathbf{K}$ of rank r can be diagonalised as follows:

$$\mathbf{K}_{m \times m} = \mathbf{P}_{m \times r} \cdot \mathbf{\Lambda}_{r \times r} \cdot \mathbf{P}_{r \times m}^{\mathrm{T}},$$

where $(\cdot)^{\mathrm{T}}$ is the transpose operation, $\mathbf{\Lambda}$ is a diagonal matrix containing the r nonzero eigenvalues of $\mathbf{K}$ in decreasing order (i.e., $\lambda_1, \ldots, \lambda_r$), and $\mathbf{P}$ is a unitary matrix that consists of the eigenvectors associated to those r eigenvalues (i.e., $\mathbf{u}_1, \ldots, \mathbf{u}_r$) constituting an orthonormal basis of $\mathbb{R}^r$. Then, the empirical kernel map is defined as:

$$\Phi_r^e : \mathbf{x}_i \to \mathbf{\Lambda}^{-1/2} \cdot \mathbf{P}^{\mathrm{T}} \cdot (k(\mathbf{x}_i, \mathbf{x}_1), \ldots, k(\mathbf{x}_i, \mathbf{x}_m))^{\mathrm{T}}. \tag{2}$$

Consider the set $\{\Phi_r^e(\mathbf{x}_1),\ldots,\Phi_r^e(\mathbf{x}_m)\}$ of the EFS images of the training points. Let $\mathbf{Z}_{m\times r}$ be the design matrix storing $\Phi_r^e(\mathbf{x}_i)$ as rows. It is easy to check that the standard dot product matrix of $\Phi_r^e(\mathbf{x}_i),\ i=1,\ldots,m$ evaluated in $\mathcal{E}$ is $\mathbf{K}$ [17], [18]. Writing $\mathbf{Z}=\mathbf{\Lambda}^{-1/2}\cdot\mathbf{P}^{\mathrm{T}}\cdot\mathbf{K}$, we obtain[1]:

$$\mathbf{Z}^{\mathrm{T}}\mathbf{Z}=\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}=\mathbf{K}.$$

Since the distances and the angles of the m vectors $\Phi(\mathbf{x}_i),(i=1,\ldots,m)$ in the feature space are uniquely determined by the dot product (i.e., $||\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_j)||^2=k(\mathbf{x}_i,\mathbf{x}_i)+k(\mathbf{x}_j,\mathbf{x}_j)-2k(\mathbf{x}_i,\mathbf{x}_j)$), the training data have the same geometrical structure in both spaces $\mathcal{F}$ and $\mathcal{E}$.

However, recall that the map Φ into the feature space is nonlinear, therefore each point in the span of the mapped input data would not necessarily be the image of some input pattern [17], [32]. This is known as the preimage problem. Obviously, this problem also appears when using the empirical kernel map, because it also corresponds to a nonlinear transformation of the input space. Note that this is not really a problem for the over-sampling of minority class, since the linear decision boundary is built in the feature space and if the classes are (almost) linearly separable in the feature space, doing local convex combination is reasonable, whether the pre-images of the synthetic points exist or not.

III. Synthetic over-sampling by convex combination in the EFS

The main hypothesis and motivation for this section is that the EFS will provide us with a more suitable class distribution for over-sampling. It is clear that when the classes are non-linearly separable (which may be the case in the input space), one should be very careful when creating synthetic patterns by convex combination of other patterns because these could lie on the majority class region. However, if the data are linearly separable (a statement that will be true if the kernel function matches the underlying learning problem), over-sampling by convex combination of patterns is not a problem. To illustrate this, consider Fig. 2 where a toy nonlinearly separable dataset have been represented by the Φ_2^e transformation using a Gaussian kernel retaining only two dominant dimensions[2].

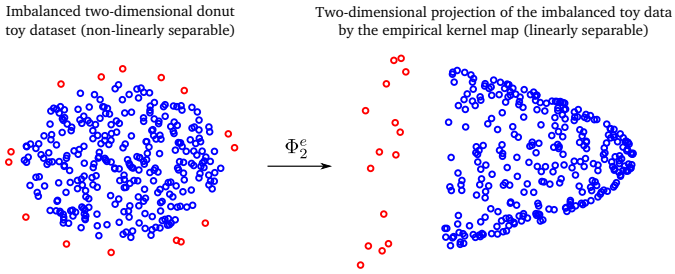


Fig. 2: Synthethic two-dimensional dataset representing a non-linearly separable classification problem and their transformation to the 2 dominant dimensions of the EFS $\mathcal{E}^{(2)}$ induced by the Gaussian kernel function (linearly separable problem).

[1]Note that $\mathbf{P}$ is a unitary matrix and $\mathbf{K}$ a symmetric matrix

[2]Dimensions associated with the highest eigenvalues of the Gram matrix.

A. Reduced empirical feature space

In this subsection, we present a reduced version of the EFS, where we select the q $(q<r)$ dominant dimensions to approximate the kernel matrix.

It has been argued in [33] that the useful information for a classification task can already be contained in a subspace of the feature space (under the assumption of smooth kernels matching the underlying learning problem). However, for kernel methods, such as SVMs, the capacity control is equivalent to some form of regularisation so that "denoising" is not necessary although it could be very useful for unregularised methods [34]. In this section, we test whether over-sampling a minority class in the reduced dimensionality EFS (as opposed to over-sampling in the full EFS) can be beneficial. One possible motivation for over-sampling in reduced dimensionality EFS may be that our over-sampling procedure relies on distances in the EFS which may become less informative as the dimensionality increases [35].

It is well-known that for any real symmetric $m\times m$ matrix $\mathbf{K}$ of rank r, we can find its real nonzero eigenvalues $\lambda_1\geq\ldots\geq\lambda_r$ and the corresponding orthonormal eigenvectors $\mathbf{u}_1,\ldots,\mathbf{u}_r$, so that $\mathbf{K}=\sum_{i=1}^r\lambda_i\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$. In this case, the best rank-q $(q<r)$ approximation to $\mathbf{K}$ is $\mathbf{K}_q=\sum_{i=1}^q\lambda_i\mathbf{u}_i\mathbf{u}_i^{\mathrm{T}}$, in the sense that it minimises $||\mathbf{K}-\mathbf{K}_q||_{\mathrm{F}}^2$ over all rank-q matrices (where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm). This concept can be said to be the main idea for the reduced EFS.

Instead of working in the full-rank EFS $\mathcal{E}$ we can operate in its lower dimensional subspace $\mathcal{E}^{(q)}$ where the kernel matrix has the form:

$$\mathbf{K}_{m\times m}^{(q)}=\mathbf{P}_{m\times q}^{(q)}\cdot\mathbf{\Lambda}_{q\times q}^{(q)}\cdot(\mathbf{P}^{(q)})_{q\times m}^{\mathrm{T}},\ q<r,$$

where $\mathbf{P}^{(q)}$ and $\mathbf{\Lambda}^{(q)}$ consist of the first q columns of $\mathbf{P}$ and $\mathbf{\Lambda}$, respectively[3].

Consider the preimage $\mathcal{F}^{(q)}$ of $\mathcal{E}^{(q)}$ under the isomorphism φ. Let $\{\mathbf{u}_j\}_{j=1}^q$ be an orthonormal basis of $\mathcal{F}^{(q)}$. Given $\mathbf{v}\in\mathcal{F}$, its projection onto $\mathcal{F}^{(q)}$ is obtained as $\{\langle\mathbf{v},\mathbf{u}_j\rangle_{\mathcal{F}}\}_{j=1}^q$. The isomorphism φ from $\mathcal{F}$ to $\mathcal{E}$ carries the structure over: $\varphi(\mathbf{v})\in\mathcal{E}$ is projected onto $\mathcal{E}^{(q)}$ as $\{\langle\varphi(\mathbf{v}),\varphi(\mathbf{u}_j)\rangle_{\mathcal{E}}\}_{j=1}^q$. Moreover, for all $j=1,\ldots,q$,

$$\langle\mathbf{v},\mathbf{u}_j\rangle_{\mathcal{F}}=\langle\varphi(\mathbf{v}),\varphi(\mathbf{u}_j)\rangle_{\mathcal{E}}.$$

Therefore, we could define the kernel associated with the reduced EFS by:

$$k^{(q)}(\mathbf{x}_i,\mathbf{x}_j)=\left\langle\Phi_q^e(\mathbf{x}_i),\Phi_q^e(\mathbf{x}_j)\right\rangle_{\mathcal{E}},$$

which, for q being the rank of $\mathbf{K}$, will correspond to k.

B. Synthetic minority over-sampling in the reduced or full-rank EFS

Once that the notion of EFS has been introduced, this subsection will show the main steps to extend a well-known over-sampling algorithm to this space.

Concerning the training phase, the first step of the proposed methodology corresponds to the computation of the training kernel matrix $\mathbf{K}$ through a predefined kernel function k. Then,

[3]We assume that the singular values are sorted.

the reduced or full-rank empirical kernel map $\Phi_q^e,\ 1 \leq q \leq r$, can be computed via the eigenvector decomposition of this training kernel matrix $\mathbf{K}$ (Eq. (2)). As said, let Z be the set generated by applying the Φ_q^e transformation to the training patterns and $\mathbf{Z}_{m\times q}$ the design matrix storing points of Z as rows. In the second step, the over-sampling process is performed over the minority class images of this $\mathbf{Z}$ matrix, resulting in the generation of n new synthetic images, arranged in the set S (and the design matrix $\mathbf{S}_{n\times q}$). More specifically, as the standard SMOTE algorithm [3] has been chosen for over-sampling, each new synthetic instance will be generated using a linear interpolation between pattern $\mathbf{x}_i$ and one of its k-nearest neighbours (both belonging to the minority class). At every step $j = 1, \ldots, n$, we create a point $\mathbf{s}_j$ in $\mathcal{E}^{(q)}$ by picking at random a minority class point $\mathbf{x}_i$ and calculating:

$$\mathbf{s}_j = \Phi_q^e(\mathbf{x}_i) + (\Phi_q^e(\hat{\mathbf{x}}_i) - \Phi_q^e(\mathbf{x}_i)) \cdot \delta,$$

where $\Phi_q^e(\hat{\mathbf{x}}_i)$ is one of the k-nearest neighbours for $\Phi_q^e(\mathbf{x}_i)$ in the EFS $\mathcal{E}^{(q)}$, and δ is a random number generated from the uniform distribution $U[0, 1]$. For simplicity, we over-sample the minority class so that the two classes become balanced. From the definition of the EFS, we know that $\varphi^{-1}(\mathbf{s}_j) \in \mathcal{F}^{(q)}$ (i.e., the representation of the new pattern in the feature space) will be unique and will lie on the line between $\varphi^{-1}(\mathbf{x}_i)$ and $\varphi^{-1}(\hat{\mathbf{x}}_i)$ (φ is a linear map). Recall that the norms and distances are preserved, e.g.:

$$\|\Phi_q^e(\hat{\mathbf{x}}_i) - \Phi_q^e(\mathbf{x}_i)\|_{\mathcal{E}} = \|\Phi(\hat{\mathbf{x}}_i) - \Phi(\mathbf{x}_i)\|_{\mathcal{F}},$$

and so are the angles, $(\Phi_q^e(\mathbf{x}_i) - \Phi_q^e(\hat{\mathbf{x}}_i))^{\mathrm{T}}(\Phi_q^e(\mathbf{x}_i) - \mathbf{s}_j) = \langle \Phi(\mathbf{x}_i) - \Phi(\hat{\mathbf{x}}_i), \Phi(\mathbf{x}_i) - \varphi^{-1}(\mathbf{s}_j)\rangle$. As a consequence, if $\Phi_q^e(\hat{\mathbf{x}}_i)$ is one of the k-nearest neighbours of $\Phi_q^e(\mathbf{x}_i)$ in the EFS, this will be so in the feature space as well.

The third step is the execution of the learning machine over the set $\varphi^{-1}(Z \cup S) \subset \mathcal{F}^{(q)}$. In this case, there are two different possibilities to consider. First, we could employ the EFS as a new representation for the data and use the classification algorithm in this new space as done in other works [36], [37]. This idea will provide us with a more easily separable and balanced space than the input space which could indeed be used for any learning machine, independently of being kernelized or not. However, when dealing with a kernel function, it could actually be more advisable to recompute the dot products between patterns (i.e., create a new over-sampled kernel matrix), due to the high number of features (the dimensionality of the EFS), which in most of the cases will increase the computational cost of the learning machine considered. To do so, synthetic samples will be used to complete the kernel matrix, by obtaining their dot product in the EFS with respect to the rest of the training patterns. Using this approach, the over- sampled training Gram matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ will be composed as follows:

$$\tilde{\mathbf{K}}^{\mathrm{tr}}_{(m+n)\times(m+n)} = \begin{pmatrix} (\mathbf{Z}\cdot\mathbf{Z}^{\mathrm{T}})_{m\times m} & (\mathbf{Z}\cdot\mathbf{S}^{\mathrm{T}})_{m\times n} \\ (\mathbf{S}\cdot\mathbf{Z}^{\mathrm{T}})_{n\times m} & (\mathbf{S}\cdot\mathbf{S}^{\mathrm{T}})_{n\times n} \end{pmatrix}. \quad (3)$$

Note that for any number of dominant dimensions q for the empirical kernel map Φ_q^e, the over-sampled kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ obtained will be positive semidefinite. Furthermore, since we are generating new patterns by a linear combination of other patterns in the dataset, the empirical kernel maps associated to $\varphi^{-1}(Z)$ and to $\varphi^{-1}(Z \cup S)$ can be said to be equivalent.

For the generalisation phase, the same steps are considered to complete the test kernel matrix, considering that the EFS images of the test patterns are derived using the same Φ_q^e map (considering only the training data). Note that in this case we will compute the dot product between train and test patterns and between test and synthetic patterns. The over- sampled test Gram matrix $\tilde{\mathbf{K}}^{\mathrm{ts}}$ will be composed as follows:

$$\tilde{\mathbf{K}}^{\mathrm{ts}}_{(m+n)\times(t)} = \begin{pmatrix} (\mathbf{Z}\cdot\mathbf{T}^{\mathrm{T}})_{m\times t} & (\mathbf{S}\cdot\mathbf{T}^{\mathrm{T}})_{n\times t} \end{pmatrix}, \quad (4)$$

where $\mathbf{T}$ is the representation in the EFS of the test patterns and t corresponds to the number of test patterns.

Note that these new over-sampled kernel matrices $\tilde{\mathbf{K}}^{\mathrm{tr}}$ and $\tilde{\mathbf{K}}^{\mathrm{ts}}$ can be used for any kernel-based algorithm.

A summary of this kernel-based over-sampling method can be seen in Fig. 3.

Algorithm synthetic over-sampling in the empirical feature space

- **Input**: Training patterns ($\mathbf{X}^{\mathrm{tr}}$), training targets (y^{tr}) and testing patterns ($\mathbf{X}^{\mathrm{ts}}$).
- **Output**: Testing targets (y^{ts})

1) Compute kernel matrix $\mathbf{K}^{\mathrm{tr}}$ for training patterns.
2) Compute the empirical kernel map Φ_q^e via $\mathbf{K}^{\mathrm{tr}}$.
3) Map training patterns to the EFS using Φ_q^e and obtain their new representation $\mathbf{Z}$.
4) Generate synthetic patterns $\mathbf{S}$ using the new representation $\mathbf{Z}$ of the training patterns.
5) Complete the over-sampled train kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ with the dot product between patterns (Eq. 3).
6) Train the learning algorithm with kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ and obtain a hyperplane $\mathbf{w}$ and a bias term b.
7) Map testing patterns to the EFS using Φ_q^e and obtain their new representation $\mathbf{T}$.
8) Complete the over-sampled test kernel matrix $\tilde{\mathbf{K}}^{\mathrm{ts}}$ with the dot product between patterns (Eq. 4).
9) Predict y^{ts} using $\tilde{\mathbf{K}}^{\mathrm{ts}}$ and the model $\{\mathbf{w}, b\}$ (Eq. 1).

Fig. 3: Different steps for the kernel over-sampling algorithm.

As mentioned before, our over-sampled points in the feature space may not have preimages in the input space. However, this does not pose a methodological problem since the class separation is formulated in the feature space.

IV. Optimising the feature space by kernel learning for over-sampling

As stated before, our first hypothesis was that over-sampling in the EFS was more advisable if the kernel function matched the underlying problem in the sense that it can asymptotically represent the function to be learned and is sufficiently smooth. In this section, we propose a method for kernel learning that would ideally provide a clearer class separation in the feature space to analyse its effect in the over-sampling method.

Ideally, we would like to find the kernel that minimises the true risk of a classifier for a specific dataset. Unfortunately, the risk is not accessible; therefore, different analytical bounds for the generalisation error have been developed in the machine learning literature with the aim of better suiting a given dataset. In the kernel machine literature, a considerable interest has been devoted to learning the "optimal" kernel given a particular classification task, as opposed to imposing them. One of the prominent approaches in kernel learning is centred kernel-target alignment (KTA) [20]. Centred KTA is data distribution

independent, making it particularly suitable for imbalanced classification. Note that KTA is related to the Fisher criterion, which maximises the distance between different classes and minimises the within class distance. This can be a useful property of the feature space in which to perform minority class over-sampling. Minority patterns would be far from the majority class region and closely clustered together.

KTA optimises the kernel by aligning it to the so-called ideal kernel matrix $\mathbf{K}_\mathrm{i}$ [19], which will submit the structure:

$$k_\mathrm{i}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 & \text{if } y_i = y_j, \\ -1 & \text{otherwise,} \end{cases}$$

where y_i is the target of pattern $\mathbf{x}_i \in X^{\mathtt{tr}}$. In this sense, $\mathbf{K}_\mathrm{i}$ will provide information about which patterns should be considered to be similar when performing a learning task.

Thus, the problem of finding an optimal kernel k is changed to the one of finding a good approximation $\mathbf{K}$ for the ideal kernel matrix $\mathbf{K}_\mathrm{i}$, given a family of kernel functions. This formulation allows to separate the optimisation from kernel machine learning and to reduce the increase in the computational cost of learning more complex kernels, given that the kernel machine will be unaffected by this higher complexity.

As said before, concerning imbalanced classification, previous studies have noted several issues in KTA for different pattern distributions [19], [38] but a recent study [20] has shown that this can be solved by the use of centred kernel matrices. The notion of centred alignment $\mathcal{A}_\mathrm{c}$ between $\mathbf{K}$ and $\mathbf{K}_\mathrm{i}$ [19], [20] is defined as:

$$\mathcal{A}_\mathrm{c}(\mathbf{K}, \mathbf{K}_\mathrm{i}) = \frac{\langle \mathbf{K}_\mathrm{c}, \mathbf{K}_{\mathrm{i}_\mathrm{c}} \rangle_\mathrm{F}}{\sqrt{\langle \mathbf{K}_\mathrm{c}, \mathbf{K}_\mathrm{c} \rangle_\mathrm{F} \langle \mathbf{K}_{\mathrm{i}_\mathrm{c}}, \mathbf{K}_{\mathrm{i}_\mathrm{c}} \rangle_\mathrm{F}}},$$

where $\mathbf{K}_\mathrm{c}$ denotes the centred version of kernel matrix $\mathbf{K}$ and is computed as:

$$\mathbf{K}_\mathrm{c} = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{m}} - \mathbf{1}_{\frac{1}{m}}\mathbf{K} + \mathbf{1}_{\frac{1}{m}}\mathbf{K}\mathbf{1}_{\frac{1}{m}},$$

being $\mathbf{1}_{\frac{1}{m}}$ a matrix with all elements equal to $\frac{1}{m}$.

Centred KTA is totally maximised when a kernel can reflect the discriminant properties of the dataset that are used to define the ideal kernel.

Consider a kernel function depending on a vector of parameters $\boldsymbol{\alpha}$. Because of the differentiability of $\mathcal{A}_\mathrm{c}$ with respect to these kernel parameters $\boldsymbol{\alpha}$, a gradient ascent algorithm can be used to maximise the alignment between the kernel matrix constructed $\mathbf{K}_{\boldsymbol{\alpha}}$ and the ideal one $\mathbf{K}_\mathrm{i}$, as follows: $\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \mathcal{A}_\mathrm{c}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_\mathrm{i})$. The alignment derivative with respect to these kernel parameters $\boldsymbol{\alpha}$ is:

$$\frac{\partial \mathcal{A}_\mathrm{c}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_\mathrm{i})}{\partial \boldsymbol{\alpha}} = \tag{5}$$
$$= \frac{1}{||\mathbf{K}_{\mathrm{i}_\mathrm{c}}||_\mathrm{F}} \left[\frac{\langle (\frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}), \mathbf{K}_{\mathrm{i}_\mathrm{c}} \rangle_\mathrm{F}}{||\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}||_\mathrm{F}} - \frac{\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{i}_\mathrm{c}} \rangle_\mathrm{F} \cdot \langle \mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}, (\frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}) \rangle_\mathrm{F}}{||\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}||_\mathrm{F}^3} \right],$$

where $||\mathbf{A}||_\mathrm{F} = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_\mathrm{F}}$ and, for arbitrary matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, it holds that $\langle \mathbf{K}_{1_\mathrm{c}}, \mathbf{K}_{2_\mathrm{c}} \rangle_\mathrm{F} = \langle \mathbf{K}_1, \mathbf{K}_{2_\mathrm{c}} \rangle_\mathrm{F} = \langle \mathbf{K}_{1_\mathrm{c}}, \mathbf{K}_2 \rangle_\mathrm{F}$ [20], which simplifies the computation.

In this paper, we will consider a generalised Gaussian kernel with covariance structure defined by a positive semidefinite matrix $\mathbf{Q}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left((\mathbf{x}_i - \mathbf{x}_j)^\mathrm{T} \mathbf{Q} (\mathbf{x}_i - \mathbf{x}_j)\right).$$

As usual, the matrix $\mathbf{Q}$ will be parametrised by $\mathbf{U}^\mathrm{T}\mathbf{U}$, where $\mathbf{U}$ is a $d \times d$ matrix (d being the dimensionality of the input space). Therefore, we can equivalently restate our problem as learning the best matrix $\mathbf{U}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left((\mathbf{x}_i - \mathbf{x}_j)^\mathrm{T} \mathbf{U}^\mathrm{T} \mathbf{U} (\mathbf{x}_i - \mathbf{x}_j)\right).$$

Now, we can compute the derivative of the kernel with respect to the entries of the $\mathbf{U}$ matrix:

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{U}}\right) = \left(\mathbf{U}(\mathbf{x}_i - \mathbf{x}_j)^\mathrm{T}(\mathbf{x}_i - \mathbf{x}_j)\right) \cdot k(\mathbf{x}_i, \mathbf{x}_j).$$

Therefore, we will optimise a vector of parameters $\boldsymbol{\alpha}$ composed of the entries of the $\mathbf{U}$ matrix.

It is important to note in this context that some attempts have been made in the literature to establish learning bounds for the Gaussian kernel with several parameters when considering large margin classifiers [39]. These studies suggest that the interaction between the margin and the complexity measure of the kernel class is multiplicative, thus discouraging the development of techniques for the optimisation of more complex and general kernels. However, recent developments have shown that this interaction is additive [40] (up to log factors), rather than multiplicative, yielding then stronger bounds. Therefore, the number of patterns needed to obtain the same estimation error with the same probability for a multi-scale kernel compared to a spherical one grows slowly (and directly depends on the number of parameters to optimise).

To demonstrate the usefulness of learning the kernels, we present in Fig. 4 a graphical representation of three two-dimensional toy datasets and their mapping Φ_2^e using a spherical Gaussian kernel with $\mathbf{Q} = 0.001 \cdot \mathbf{I}_d$, an optimised spherical Gaussian kernel obtained through centred KTA and an optimised generalised Gaussian kernel.

Summarising, kernel learning will be applied before the over-sampling procedure to learn a suitable kernel $\mathbf{K}_{\boldsymbol{\alpha}^*}$ for the data representation. After this, the EFS Φ_q^e associated to this kernel $\mathbf{K}_{\boldsymbol{\alpha}^*}$ will be computed, and then, the images of the training patterns for the minority class (contained in the $\mathbf{Z}$ matrix) will be over-sampled. For comparison purposes, we will also test the optimization of a spherical Gaussian kernel with one kernel parameter via kernel-target alignment.

V. Unified framework for preferential over-sampling

As stated before, several approaches have been developed in the literature for handling imbalanced data, and a large number of these contributions are based on analysing the patterns which could be more suitable for over-sampling, giving rise to approaches based on over-sampling on the class boundary [5], [7] or in the within class "safe region" [6] (these techniques are commonly referred to as weighted over-sampling). However, to our best knowledge, there is no principled method for choosing the region of the minority class to be used for over-sampling. In this section we propose a new adaptive weighted over-sampling technique that naturally spans unweighted and weighted over-sampling methods (both on the boundary and within class). To do so, our approach will take advantage of the spatial distribution of the patterns according to the optimal hyperplane obtained from the SVM solution.

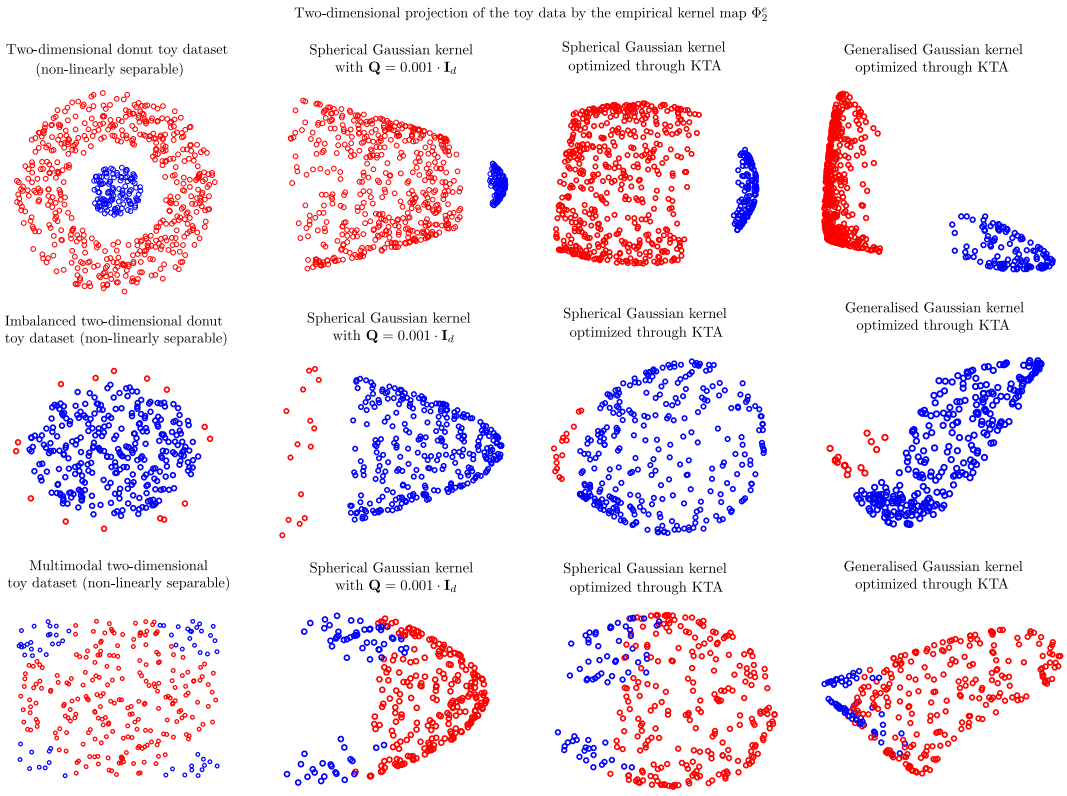


Fig. 4: Synthethic two-dimensional datasets representing non-linearly separable classification problems and their transformation to the 2 dominant dimensions of the EFS induced by the Gaussian kernel function (linearly separable problem).

A. *Knowledge extraction: Spatial distribution of the patterns*

Weighted over-sampling techniques are based on the idea that not all the patterns of the dataset are equally important and suitable for over-sampling and therefore, they should not contribute equally to the new synthetic data. One of the first steps of these methodologies corresponds to the identification of the '*useful*' patterns to be used for over-sampling. Most of the approaches in the literature do so by analysing local neighbourhood of points in the minority class. In this paper, however, we will derive a weighted over-sampling technique considering the spatial distribution of the patterns with respect to the optimal SVM hyperplane. In particular, the patterns to be used for over-sampling will be selected based on their position and distance to the optimal hyperplane.

However, as stated before, the soft-margin optimisation of the SVM paradigm poses a serious problem for imbalanced datasets. Therefore, for the purpose of weighted over-sampling, we use the cost-sensitive approach giving more importance to errors committed by patterns belonging to the minority class [23]. The cost-sensitive SVM approach consists of introducing different penalty factors C_{+1} and C_{-1} for the positive and negative SVM slack variables during training. The primal SVM problem is transformed into:

$$\frac{1}{2}||\mathbf{w}||^2 + C_{+1} \sum_{\{i|y_i=+1\}} \xi_i + C_{-1} \sum_{\{i|y_i=-1\}} \xi_i,$$

subject to the constraints:

$$y_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall_i \in \{1, \ldots, m\}.$$

For simplicity, we will set $C_{+1} = \frac{m_{-1}}{m_{+1}} \cdot C_{-1}$, where $+1$ is assumed to be the minority class, m_{+1} is the number of patterns belonging to class $+1$ and m_{-1} the number of patterns belonging to class -1. The ratio $\frac{m_{-1}}{m_{+1}}$ is usually known as the imbalanced ratio.

As stated before, each synthetically generated point $\mathbf{s}_z \in \mathcal{E}^{(q)}$, $z = 1, \ldots, n$, in the minority class represented by training samples X_+^{tr} is generated by first picking a pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ from X_+^{tr} and then constructing their convex combination in the EFS $\mathcal{E}^{(q)}$:

$$\mathbf{s}_z = \Phi_q^e(\mathbf{x}_i) + (\Phi_q^e(\mathbf{x}_j) - \Phi_q^e(\mathbf{x}_i)) \cdot \delta,$$

where δ is a random number generated from the uniform distribution $U[0, 1]$.

B. *Optimisation of the over-sampling procedure*

The points $\mathbf{x}_i$ and $\mathbf{x}_j$ will be randomly selected based on their relative position in the feature space with respect to the separating hyperplane. Because the norm of $\mathbf{w}$ is 1, the signed distance of $\Phi(\mathbf{x}_i) \in \mathcal{F}^{(q)}$ from the hyperplane is given by $f(\mathbf{x}_i) = \mathbf{w} \cdot \Phi(\mathbf{x}_i) + b$. Note that if $\Phi(\mathbf{x}_i)$ is on the '*right*' side of the hyperplane $f(\mathbf{x}_i)$ is positive, otherwise it is negative[4]. We will represent the selection process as draws from a

[4]If $\Phi(\mathbf{x}_i)$ lies on the separating hyperplane, then $f(\mathbf{x}_i) = 0$.

multinomial distribution over X_+^{tr} (i.e., patterns belonging to the minority class) with natural parameters $\mu_i = -\beta \cdot f(\mathbf{x}_i)$, where $\beta \in \mathbb{R}$ is a scale parameter. Using the soft-max link function, the probability of picking $\mathbf{x}_i \in X_+^{\text{tr}}$ is:

$$P(\mathbf{x}_i) = \frac{\exp(-\beta f(\mathbf{x}_i))}{\sum_{\mathbf{x} \in \mathbf{X}_+^{\text{tr}}} \exp(-\beta f(\mathbf{x}))}. \tag{6}$$

Note that when $\beta < 0$, points deep within the minority class (in the feature space) are more likely to be picked; when $\beta > 0$, points closer to the class boundary or lying inside the opposite class are preferred and, when $\beta = 0$, all the points are equally likely to be chosen, as this will correspond to the uniform distribution over X_+^{tr}. This approach naturally spans different approaches to weighted [5]–[7] and unweighted [3] over-sampling previously introduced in the literature.

For selecting the pairs $(\mathbf{x}_i, \mathbf{x}_j) \in X_+^{\text{tr}}$ we could use two different ideas:

- Pick $\mathbf{x}_i$ and $\mathbf{x}_j$ independently with respect to the distribution of Eq. (6).
- Pick $\mathbf{x}_i$ according to the distribution of Eq. (6) and select $\mathbf{x}_j$ using k-nearest neighbours method [30].

In most of the weighted approaches in the literature they make use of the k-nearest neighbours method because they obtain the spatial distribution information of the patterns according to their neighbourhood. However, for this approach, note that it is actually more advisable to select $\mathbf{x}_i$ and $\mathbf{x}_j$ independently according to the probability distribution obtained, because otherwise the effect of the preferential learning in the over-sampling process could be smoothed (i.e., picking points by the k-nearest neighbours approach may differ to a large extent to the selection made with the probability function).

Based on the arguments in Section III, over-sampling of the minority class in the feature space is done through over-sampling in the EFS. Note that the patterns preferred for over-sampling in the input space could not be the ones preferred in the feature space, therefore the use of the EFS is needed for this methodology as well.

To optimise the β values (as different β values will induce different synthetic patterns), we will test two approaches:

- The first idea is to use a single value of β found by, e.g., cross-validation over a set of p predefined β values.
- The second idea is to use multiple β values within the framework of multiple kernel learning (MKL), i.e., a combination of different over-sampled kernel matrices. For a particular value of β, we denote by $\tilde{\mathbf{K}}_\beta$ the kernel matrix obtained on the extended data sample (i.e., including over-sampled points obtained using β). We fix a set of β values $\{\beta_1, \ldots, \beta_p\}$ and compute the over-sampled kernel matrices $\{\tilde{\mathbf{K}}_{\beta_1}, \ldots, \tilde{\mathbf{K}}_{\beta_p}\}$. Then, using KTA, we could derive a kernel matrix $\mathbf{K}_{\boldsymbol{\omega}} = \sum_{k=1}^{p} \omega_k \tilde{\mathbf{K}}_{\beta_k}$ with $\omega_k \geq 0$ and $\sum_{k=1}^{p} \omega_k = 1$ (convex combination of kernel matrices $\tilde{\mathbf{K}}_{\beta_k}$) by multiple kernel learning techniques. Thus, this strategy will be more flexible than the cross-validation one, because we can optimise a combination of over-sampled kernel matrices, instead of restricting the solution to only choosing the best performing one. For the optimisation we will need to define an extended ideal kernel matrix $\tilde{\mathbf{K}}_{\text{i}}$, by introducing the information of the new synthetic patterns (recall that all these patterns will belong to the minority class). The optimisation problem to solve in this case will be the following:

$$\max_{\boldsymbol{\omega} \in \mathcal{M}} \frac{\left\langle \tilde{\mathbf{K}}_{\boldsymbol{\omega}_c}, \tilde{\mathbf{K}}_{\text{i}} \right\rangle_{\text{F}}}{||\tilde{\mathbf{K}}_{\boldsymbol{\omega}_c}||_{\text{F}}},$$

where $\mathcal{M} = \{\boldsymbol{\omega} : ||\boldsymbol{\omega}||_2 = 1\}$. Note that since we are trying to align the real kernel matrix $\tilde{\mathbf{K}}$ with the ideal one $\tilde{\mathbf{K}}_{\text{i}}$ the value of $\left\langle \tilde{\mathbf{K}}_{\text{i}}, \tilde{\mathbf{K}}_{\text{i}} \right\rangle_{\text{F}}$ does not change and it can be obviated in the optimisation process. The Quadratic Programming (QP) optimization problem associated can be seen in [20].

Fig. 5 shows the representation of the training data for the cleveland0vs4 dataset in different EFS using the transformation Φ_2^e (original EFS, over-sampled EFS for $\beta = -5$ and $\beta = 5$, and optimised over-sampling through MKL). In this case, the difference between over-sampling for different β values could be difficult to appreciate. However, for the case of the optimised over-sampled EFS one can note that the class separation increases and the within class decreases (recall that KTA was related to the Fisher criterion).

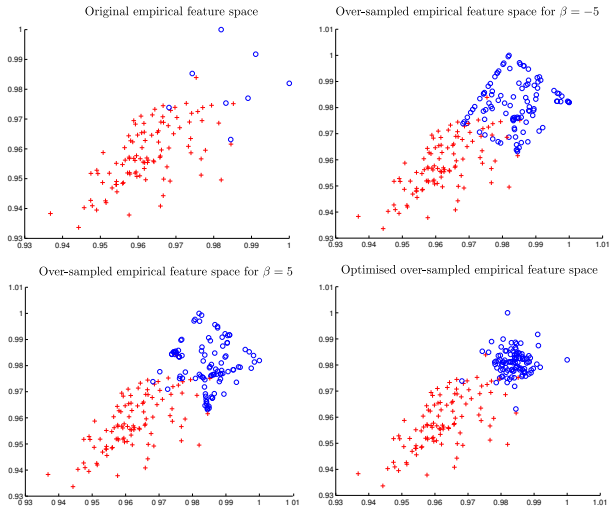


Fig. 5: Empirical feature spaces for the cleveland0vs4 dataset associated to the original data, over-sampling for different β values and optimised over-sampling.

In the same vein, Fig. 6 shows the case of the training data for the led7digit02456789vs1 dataset and the transformation Φ_2^e. In this case, the difference for the over-sampling procedure when using different β values can be easily appreciated.

VI. Experimental results

The proposed methodologies have been tested considering Support Vector Machines (SVM) [14] and the well-known SMOTE algorithm [3]. 50 binary benchmark datasets from the UCI repository with different imbalance ratios (proportion of majority patterns with respect to minority ones) have been used for the analysis to test the performance of the methods in different situations. The characteristics of these datasets can

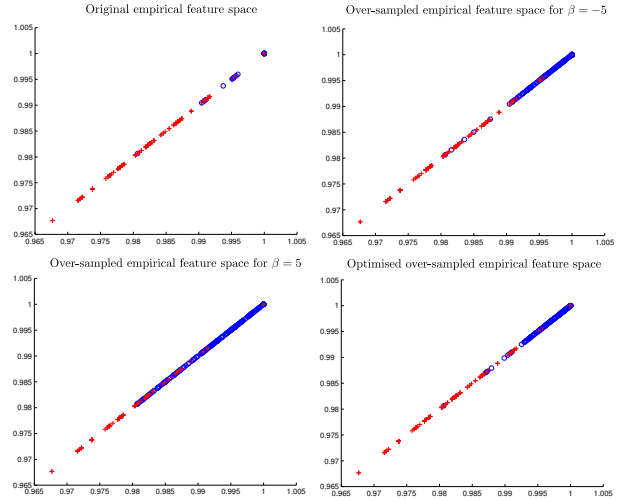


Fig. 6: Empirical feature spaces for the led7digit02456789vs1 dataset associated to the original data, over-sampling for different β values and optimised over-sampling.

be seen in TABLE I. As done in other over-sampling state-of-the-art works [10], some multiclass datasets have also been considered by grouping some classes, e.g. ecoli1 represents the ecoli dataset when considering class 1 versus the rest, and yeast0359vs78 is the yeast dataset when grouping classes 0, 3, 5, and 9 versus classes 7 and 8 in order to obtain higher imbalance ratio (IR) values.

A stratified 5×2-fold Dietterich technique was performed to divide the data and the results are taken as mean and standard deviation of the selected measures as done elsewhere (e.g. [10]). Each experiment over each data partition has been repeated 6 times using a different seed to obtain more robust results[5] (i.e., at the end of the execution we will have 30 results for each dataset). The Gaussian kernel was used. The kernel width and the cost parameter of SVM were selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ by means of a nested 5-fold method applied to the training set. As done in other works [8], [9], the number of synthetic patterns generated was that needed to balance the distributions, i.e. after applying the over-sampling process, the number of majority and minority patterns were the same. $k = 3$ nearest neighbours were evaluated to generate synthetic samples, in order to minimise the chance that synthetic patterns are generated in the majority class region when using the standard SMOTE technique.

The results have been reported in terms of two metrics, one of them specially designed to deal with imbalanced data:

1) The well-known Accuracy metric (Acc), which corresponds to the ratio of correctly classified patterns and measures overall performance. For the case of imbalanced datasets, it is important to note that this metric may not be the best option, since the classification of the minority class may be compromised for the sake of the majority one (it does not distinguish between the numbers of correctly classified examples of each class), and we could therefore obtain a trivial classifier always outputting the majority class.
2) The Geometric Mean of the sensitivities ($GM = \sqrt{S_{\mathrm{p}} \cdot S_{\mathrm{n}}}$), where S_{p} is the sensitivity for the positive class (ratio of correctly classified patterns considering only this class) and S_{n} is the sensitivity for the negative one.

[5]Recall that synthetic patterns are randomly generated.

TABLE I: Datasets used for the experiments (N corresponds to the total number of patterns, d to the dimensionality of the input space and IR to the imbalance ratio).

Dataset	N	d	IR	Dataset	N	d	IR
ecoli0vs1	352	7	1.84	ecoli067vs35	354	7	9.41
glass1	342	9	1.85	glass04vs5	146	9	9.43
wisconsin	1092	9	1.86	ecoli0267vs35	358	7	9.53
pima	1228	8	1.87	yeast05679vs4	844	8	9.55
yeast1	2374	8	2.46	ecoli067vs5	352	6	10.00
haberman	488	3	2.81	glass016vs2	306	9	10.77
vehicle2	1352	18	2.89	ecoli01vs5	384	6	11.00
vehicle1	1352	18	2.91	led7digit02456789vs1	708	7	11.21
vehicle3	1352	18	3.00	glass06vs5	172	9	11.29
vehicle0	1352	18	3.25	glass0146vs2	328	9	11.62
glass0123vs456	342	9	3.28	glass2	342	9	12.15
ecoli1	536	7	3.39	ecoli0147vs56	530	6	12.25
newthyroid1	344	5	5.14	cleveland0vs4	276	13	12.80
newthyroid2	344	5	5.14	ecoli0146vs5	448	6	13.00
ecoli2	536	7	5.54	shuttle0vs4	2926	9	13.78
yeast3	2374	8	8.13	yeast1vs7	734	7	14.29
ecoli3	536	7	8.57	ecoli4	536	7	15.75
ecoli034vs5	320	7	9.00	pageblocks13vs4	754	10	16.14
yeast0359vs78	808	8	9.10	abalone9-18	1168	10	16.70
ecoli046vs5	324	6	9.13	glass016vs5	294	9	20.00
yeast0256vs3789	1606	8	9.16	yeast2vs8	770	8	23.06
yeast02579vs368	1606	8	9.16	shuttle2vs4	206	9	24.75
ecoli0347vs56	410	7	9.25	yeast4	2374	8	28.68
ecoli01vs235	390	7	9.26	yeast5	2374	8	32.91
yeast2vs4	822	8	9.28	yeast6	2374	8	41.39

The measure considered during the hyperparameter selection was GM, given its robustness for imbalanced datasets.

The source codes in Matlab for the methods developed in this paper are available, together with the datasets, partitions and the results on the website associated with this paper[6].

The purpose of this section is three-fold. The first experiment is intended to test whether the empirical kernel map provides a more suitable space for over-sampling than the input space when dealing with kernel methods and analyses the effect of the number of dimensions chosen for over-sampling (i.e., the influence of the concentration of spectral properties). The second experimental subsection will complement the approach proposing a new kernel learning algorithm, to optimise a more flexible kernel function, which would ideally better fit the data. The purpose of this experiment is to test whether the kernel function chosen influences the results and how, optimising this kernel function the synthetic generated data will be better adapted to the classification problem. Finally, the third experiment focuses on the case of weighted or preferential over-sampling to analyse which patterns should be more prone to be over-sampled and test a new multiple kernel learning algorithm for optimising the generated patterns.

TABLE II contains information about all of the methods used for this three-fold experimentation and a brief summary (mean and standard deviation) of the mean results obtained along the 50 datasets used for the experimentation.

[6]http://www.uco.es/grupos/ayrna/efso

The complete set of results for all of the methods can be seen in the webpage associated to this paper[6], including the individual results for all the different datasets. For the sake of comparison, we included the results obtained by a majority class rule (MCR) classifier as a baseline result for the problem (i.e., a naïve rule that classify all the patterns as belonging to the majority class). From the results of MCR it can be seen that Acc is not a proper metric to take into account, since this trivial methodology achieves the best results in some cases (haberman, yeast05679vs4, glass016vs2, glass0146vs2, glass2, yeast4 and yeast6). In the following subsections, we will perform three differentiated statistical tests to validate the previously stated hypotheses.

TABLE II: Abbreviation for all the methods considered for the experimentation and mean and standard deviation results (Mean_{SD}) for all of the datasets.

Algorithm	$Acc(\%)$	$GM(\%)$
Majority class rule classifier (MCR)	$86.70_{0.53}$	$0.00_{0.00}$
SVM without over-sampling (SVM)	$\mathbf{93.35_{2.02}}$	$77.28_{12.07}$
SVM applying over-sampling in the input space (OIS)	$90.33_{3.11}$	$85.72_{8.50}$
SVM with over-sampling in the empirical feature space ($OEFS$)	$90.24_{3.50}$	$86.30_{8.14}$
SVM with over-sampling in the reduced empirical feature space ($OREFS$)	$90.41_{3.27}$	$86.83_{7.20}$
SVM with an optimised spherical kernel for over-sampling (OSK)	$\mathit{90.95_{3.16}}$	$80.45_{11.34}$
SVM with an optimised generalised kernel for over-sampling (OGK)	$89.45_{4.09}$	$\mathit{87.17_{6.86}}$
SVM with over-sampling via cross-validated preferential learning ($OCPL$)	$90.15_{3.35}$	$\mathbf{87.18_{6.74}}$
SVM with over-sampling via preferential multiple kernel learning ($OPMKL$)	$90.59_{3.45}$	$86.89_{7.20}$

The best method is in **bold** face and the second one in *italics*

A. First experiment: Over-sampling in the EFS

In this subsection, we will validate the hypothesis that the EFS is a more suitable space for over-sampling than the input space. Furthermore, we will test whether by optimising the dimensionality of this space the generated patterns are more adequate for the classification problem. To do so, we will test four different approaches: SVM, OIS, $OEFS$ and $OREFS$ (see TABLE II for the meaning of the acronyms).

As said before, we discarded all dimensions that correspond to zero eigenvalues for the computation of the EFS for $OEFS$. Furthermore, we performed a nested 5-fold cross-validation over the training sets of the number of dominant dimensions for all the datasets considered when considering over-sampling in the reduced EFS ($OREFS$). To do so, we considered the following values for the q value of the empirical kernel map Φ_q^e: $q \in \{\lfloor 0.1r \rfloor, \lfloor 0.25r \rfloor, \lfloor 0.5r \rfloor, \lfloor 0.75r \rfloor, r\}$, where r is the original rank of the training kernel matrix $\mathbf{K}$ and $\lfloor \cdot \rfloor$ is the floor function.

It can be seen that the results in GM for SVM are in general very poor (analyse for example the case of the haberman and glass2 datasets). Concerning the OIS method, it can be seen that in some cases the results of $OEFS$ are much better (analyse the result of the glass04vs5 dataset where SVM even obtained better results or the case of the glass016vs5 dataset). In relation to the effect of controlling the dimensionality, it can be seen that $OREFS$ generally yielded similar or better performance than $OEFS$ (see the result of the yeast2vs8 and led7digit02456789vs1 datasets, two examples which will be afterwards analysed). When taking Acc into account, it can be seen that the three over-sampling methods obtain very similar values (although $OEFS$ and $OREFS$ obtain better results in some cases, e.g. ecoli0267vs35).

TABLE III shows the test mean rankings (1 for the best method and 4 for the worst) for the methods considered in this experiment along all of the 50 datasets in terms of Acc and GM. The results show that SVM is the best performing method for Acc but the worst performing when considering a metric that takes into account the imbalanced nature of the data (GM). Furthermore, it is shown that both approaches for over-sampling in the EFS ($OEFS$ and $OREFS$) outperfomed the results obtained when over-sampling in the input space (OIS). Finally, it can be seen that controlling the EFS dimensionality we improve the results in most cases, as the $OREFS$ method obtained better mean results than $OEFS$.

To quantify whether a statistical difference exists among the algorithms compared, a procedure is employed to compare multiple classifiers in multiple datasets [41]. TABLE III also shows the result of applying the non-parametric statistical Friedman's test (for a significance level of $\alpha = 0.05$) to the mean Acc and GM rankings. It can be seen that the test rejects the null-hypothesis that all of the algorithms perform similarly in mean ranking for both metrics (note that for GM the significant differences are larger).

TABLE III: Mean ranking results for SVM, OIS, $OEFS$ and $OREFS$.

Ranking	SVM	OIS	$OEFS$	$OREFS$
Acc	**1.53**	3.21	2.74	*2.52*
GM	3.64	2.61	*1.96*	**1.79**
Friedman's test				
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 2.66)$				
F-value$_{Acc}$: $21.06 \notin C_0$, F-value$_{GM}$: $35.70 \notin C_0$				

On the basis of this rejection and following the guidelines of [41], we consider the best performing methods in GM (the two proposals, $OEFS$ and $OREFS$) as control methods for the post-hoc test and we compare them to the rest according to their rankings. It has been noted that the approach of comparing all classifiers to each other in a post-hoc test is not as sensitive as the approach of comparing all classifiers to a given classifier (control method). One approach to this latter type of comparison is the Holm's test. The test statistics for comparing the i-th and j-th method using this procedure is:

$$z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}},$$

where k is the number of algorithms, N is the number of datasets and R_i is the mean ranking of the i-th method. The z value is used to find the corresponding probability from the table of normal distribution, which is then compared with an appropriate level of significance α. Holm's test adjusts the value for α in order to compensate for multiple comparisons.

This is done in a step-up procedure that sequentially tests the hypotheses ordered by their significance. We will denote the ordered p-values by $p_1, p_2, \ldots, p_k$ so that $p_1 \leq p_2 \leq \ldots \leq p_k$. Holm's test compares each p_i with $\alpha^*_{\text{Holm}} = \alpha/(k-i)$, starting from the most significant p value. If p_1 is below $\alpha/(k-1)$, the corresponding hypothesis is rejected and we allow to compare p_2 with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well.

To analyse the results obtained from the Holm's test see TABLE IV. For the $OEFS$ method, the test concluded that there were statistically significant differences with SVM for Acc (note that in this case SVM obtained better results), SVM for GM and OIS for GM as well. This indicates that, although $OEFS$ obtained worst results for Acc in comparison with SVM, the results for GM are significantly better with comparison to SVM and OIS (therefore proving the fact that the EFS provides a more suitable space for over-sampling by convex combination of patterns). Concerning the $OREFS$ method, the same results are obtained, but there are also significant differences when considering the OIS method for Acc, which could indicate that over-sampling in the empirical feature space can be beneficial with other purposes, for example, for ensuring the class boundaries.

TABLE IV: Results of the Holm procedure using $OEFS$ and $OREFS$ as control methods (CMs) when compared to SVM and OIS: corrected α values, compared method and p-values, all of them ordered by the number of comparison (i).

CM: $OEFS$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.016	SVM	0.0000_{--}	SVM	0.0000_{++}
2	0.025	OIS	0.0687	OIS	0.0118_{++}
3	0.050	$OREFS$	0.3941	$OREFS$	0.5102
CM: $OREFS$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.016	SVM	0.0000_{--}	SVM	0.0000_{++}
2	0.025	OIS	0.0007_{++}	OIS	0.0014_{++}
3	0.050	$OEFS$	0.3941	$OEFS$	0.5102

Win $(++)$ or lose $(--)$ with statistical significant difference for $\alpha = 0.05$

In relation to the optimal value of the dimensionality of the EFS, it can be said that the decay rate of the eigenvalues is related to the smoothness of the kernel function and the actual number of necessary dimensions depends on the interplay between the kernel and the learning dataset. In this case, the mean value obtained from the cross-validation step for the number of dimensions was $(0.42 \pm 0.29)r$. More specifically, Fig. 7 shows the histogram of the optimal dimensionality of the EFS for all the datasets tested, where it can be seen that in most of the cases $\lfloor 0.5r \rfloor$ is enough to contain all the relevant information about the dataset.

As said before, one of the hypothesis for controlling the dimensionality of the EFS was that our over-sampling algorithm relies on distances computed in the EFS, which may become less informative as the EFS dimensionality increases. Fig. 8 shows the histogram of distances between pairs of patterns for different values of the dimensionality of the EFS ($\lfloor 0.1r \rfloor$ and $1r$) for two datasets where the $OREFS$ method obtained much better results than $OEFS$ and where this spectral properties phenomenon can be appreciated. Note that for the case of the yeast2vs8 dataset, using all of the dimensions ($1r$) corresponds to over-sampling in an almost randomly fashion as the k-nearest neighbours rule will not be very precise since most of the distances between pair of patterns are similar.

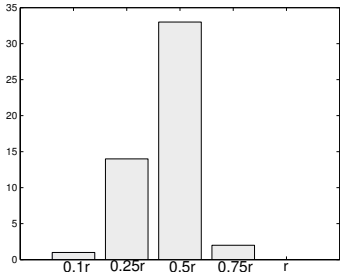


Fig. 7: Histogram of the mean optimal dimensionality of the EFS for all datasets. The abscissa axis represents the mean value, over the 30 results, for the rate of the rank of the kernel matrix. The ordinate axis shows the number of datasets where this value was selected from the cross-validation step.

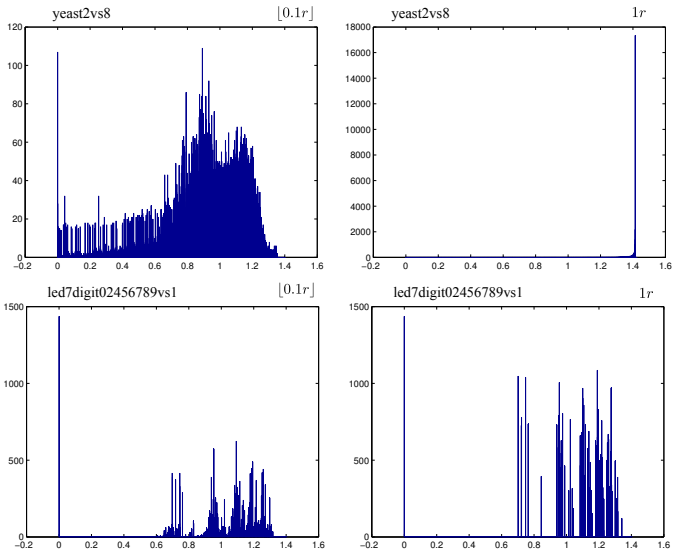


Fig. 8: Histogram of distances between pair of patterns for different dimensionality values of the EFS. The abscissa axis represents the distance between two patterns and the ordinate axis the occurrence of each distance.

From the results obtained in this subsection, several conclusions can be drawn. First, it can be stated that over-sampling by convex combination of patterns is more suitable in an (ideally) linearly separable space such as the EFS. In this sense, the method is able to obtain better results in metrics that take into account the imbalanced nature of a dataset without compromising the overall accuracy. However, over-sampling in the input space do not seem to achieve this balance between these two metrics, this fact indicating that when using a convex combination of patterns in a possibly nonlinearly separable space we could generate patterns in unwished areas of the input space. Concerning the optimisation of the number of

dominant dimensions for the feature space this methodology seems to improve the results in some cases, thus encouraging further development of an analytical methodology to perform this selection.

B. Second experiment: Influence of the kernel function

For this experiment we compare three different proposals: firstly, $OEFS$, which will be used as a baseline method to test if the optimisation of the kernel function leads to better results, secondly, SVM with an optimised spherical Gaussian kernel (the same kernel than for $OEFS$ but optimised through KTA) for performing the over-sampling in the empirical feature space (OSK) and finally SVM with an optimised generalised Gaussian kernel in the empirical feature space (OGK).

In this work, the iRprop$^+$ algorithm is used to optimise the aforementioned centred KTA, because of its robustness [42]. As stated before, the optimisation of the gradient based methods is guaranteed only to find a local minimum; therefore, the quality of the solution can be sensitive to initialisation. The gradient norm stopping criterion was set to 10^{-5} and the maximum number of conjugate gradient steps to 10^2 [42]. For the optimisation, we also include a γ parameter as the kernel width in the generalised Gaussian kernel, which will indeed do the parameters initialisation easier. The initial point for γ for all of the methods tested was chosen from the set $\{10^{-1}, 10^0, 10^1\}$, analysing the best result in alignment for the three values. The $\mathbf{Q}$ matrix for the generalised Gaussian kernel is initialised as the Moore-Penrose pseudoinverse of the covariance of the training points: $\mathbf{Q} = (cov(\mathbf{X}^{\mathrm{tr}}))^+$, to address the problem of ill-conditioned covariance matrices. Once the kernel has been optimised via KTA, we optimise the C parameter using cross-validation within the values $\{10^{-3}, 10^{-2}, \ldots, 10^3\}$ (this two stage optimisation method is also referred in the literature as second-order method [43]).

From the results (that can be found in the website[6]) one can see that OSK and OGK obtained in some cases even better results in Acc than SVM, this fact could be due to the application of the kernel optimisation through KTA, which selected a more optimal kernel than the cross-validation method. Analysing GM it can be seen that the performance of the spherical Gaussian kernel is not satisfactory. In optimising the spherical kernel, a cross-validation methodology should be preferred to KTA. To see this, analyse the case of the yeast0359vs78 and glass016vs2 datasets, where although OSK incorporates a over-sampling stage, SVM obtained better GM results. Finally, it can be seen that OGK yielded a much better performance in most of the cases (analyse the shuttle0vs4 dataset), demonstrating therefore that a more flexible kernel combined with kernel learning techniques could optimise the separation of the classes in the feature space, a necessary condition for over-sampling by convex combination of patterns.

As done before, TABLE V shows the mean ranking results for the three methods considered in this subsection and the result of applying the non-parametric Friedman's test (the test accepted the null-hypothesis that all of the algorithms perform similarly for Acc and rejected it for GM). From the results obtained it can be seen that when using a spherical Gaussian kernel, as in $OEFS$ (optimised through cross-validation) and OSK (optimised by KTA), the results are comparable and the methods obtain very similar mean ranking results. In this case, it is clear that the cross-validation method obtains better GM results as this is the metric used for the parameters selection stage. However, when using a more flexible kernel, such as the one considered in the OGK method, the results can be significantly improved. Note that applying cross-validation to the generalised kernel could possibly improve GM results, but the computational task required would be infeasible.

TABLE V: Mean ranking results for $OEFS$, OSK and OGK.

Ranking	$OEFS$	OSK	OGK
Acc	*1.94*	**1.86**	2.20
GM	*2.15*	2.37	**1.48**
Friedman's test			
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 3.09)$			
F-value$_{Acc}$: $1.60 \in C_0$, F-value$_{GM}$: $13.41 \notin C_0$			

On the basis of Friedman's test rejection, the Holm test for multiple comparisons has been applied (see TABLE VI), and the test concluded that there were statistically significant differences for GM both when considering OSK and $OEFS$. As stated before, there were no statistically significant differences for Acc.

TABLE VI: Results of the Holm procedure using OGK as the control method when compared to OSK and $OEFS$: corrected α values, compared method and p-values, all of them ordered by the number of comparison (i).

CM: OGK		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.025	OSK	–	OSK	0.0000_{++}
2	0.050	$OEFS$	–	$OEFS$	0.0008_{++}

Win ($++$) or lose ($--$) with statistical significant difference for $\alpha = 0.05$

The results in this subsection show that over-sampling in the EFS is affected by the kernel function (although spherical Gaussian kernel has been proven to show promising results in the previous subsection), kernel selection/learning which is indeed a complex issue, shows (much) better results when employing a more flexible kernel such as the one used. Therefore, different kernel learning techniques could be explored in the future for the purpose of over-sampling in the EFS.

C. Third experiment: Preferential over-sampling

This experimental subsection is intended to test if there are patterns which are more suitable for over-sampling and if a general adaptive approach, yielding solutions based on unweighted over-sampling, borderline weighted over-sampling or '*safe*' level weighted over-sampling, could achieve better results than standard unweighted over-sampling. To do so, we compare $OEFS$ to two different approaches: the first one based on a cross-validation strategy ($OCPL$) and the second one based on kernel learning techniques ($OPMKL$).

As said before, to test this idea, we first obtain the spatial distribution of the patterns based on a cost-sensitive SVM hyperplane and we use a parametrised soft-max link function (Eq. (6)) to assign different probabilities of being over-sampled to the patterns according to this spatial distribution. This parametrisation is made using a β scale parameter, which will be optimised through cross-validation ($OCPL$) within a set of values and through kernel learning techniques ($OPMKL$). For the experiments, we select the set $\beta \in \{-5, -1, 0, 1, 5\}$.

Analysing the results obtained it can be seen that both $OCPL$ and $OPMKL$ obtain very competitive results both for Acc and GM. For some cases, the results obtained are equal since $OPMKL$ also includes the solutions of $OCPL$.

Once again, TABLE VII shows the mean ranking results when comparing these two approaches to the standard proposed technique $OEFS$. In this case, the Friedman's test accepted the null-hypothesis that the algorithms perform similarly for Acc and rejected it for GM. From these results, it can be seen that both methods outperform the standard proposal or at least yield similar performance (when considering Acc).

TABLE VII: Mean ranking results obtained by $OEFS$, $OCPL$ and $OPMKL$.

Ranking	$OEFS$	$OCPL$	$OPMKL$
Acc	*1.94*	2.13	**1.93**
GM	2.48	*1.93*	**1.59**
Friedman's test			
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 3.09)$			
F-value$_{Acc}$: $0.63 \in C_0$, F-value$_{GM}$: $12.38 \notin C_0$			

The Holm's test for multiple comparisons has been also applied (see TABLE VIII). For both approaches ($OCPL$ and $OPMKL$), the test concluded that there are statistically significant differences for GM when compared to $OEFS$, indicating that preferential over-sampling is preferable to the uniform one [6]. Furthermore, although the cross-validation strategy obtains very good results, the multiple kernel strategy seems to be more general and yields a slightly better performance (in this case, there are statistically significant differences for $\alpha = 0.10$).

TABLE VIII: Results of the Holm procedure using $OCPL$ and $OPMKL$ as control methods when compared to other state-of-the-art methods: corrected α values, compared method and p-values, ordered by the number of comparison (i).

CM: $OPMKL$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.025	$OEFS$	–	$OEFS$	0.0000_{++}
2	0.050	$OCPL$	–	$OCPL$	0.0891_{+}
CM: $OCPL$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.025	$OEFS$	–	$OEFS$	0.0059_{++}
2	0.050	$OPMKL$	–	$OPMKL$	0.0891_{-}

Win $(++)$ or lose $(--)$ with statistical significant difference for $\alpha = 0.05$
Win $(+)$ or lose $(-)$ with statistical significant difference for $\alpha = 0.10$

To analyse the most appropriate region for over-sampling we analyse the optimal β values obtained from the cross-validation (see Fig. 9 for a histogram of the values). Recall that when $\beta < 0$, points within the minority class (in the feature space) are more likely to be picked; when $\beta > 0$ points on the class boundary or even on the other side of the hyperplane are preferred and when $\beta = 0$ all the points are equally likely to be chosen. It can be seen that for most datasets, over-sampling within "interior" of the minority class is preferable.

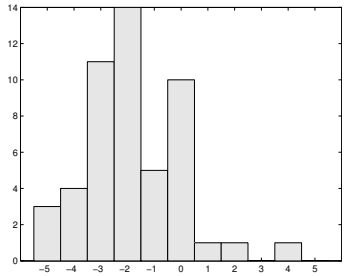


Fig. 9: Histogram of the mean values for the beta parameter used in the over-sampling process. The x coordinate represents the different mean β values and the y the number of datasets where the value was selected from the cross-validation process.

VII. Conclusions

This paper explores the notion of over-sampling in the feature space induced by a kernel function to deal with imbalanced classification problems. Since the feature space is not directly accessible, the empirical feature space is used instead (a Euclidean space that preserves the structure of the original feature space). Over-sampling is tackled by convex combination of patterns (as usually done in the state-of-the-art) and we focus on the paradigm of kernel methods. We explore the ideas of over-sampling in the full and reduced-rank empirical feature space, the optimisation of the feature space by kernel learning techniques and the notion of preferential over-sampling which analyses which patterns should be more prone to be over-sampled. From the results of a thorough set of experiments over 50 imbalanced datasets, several conclusions can be drawn: firstly, over-sampling in the empirical feature space is seen to yield better performance than over-sampling in the input space; secondly, the control of the dimensionality of the empirical feature space could lead to better results due to the concentration of spectral properties; thirdly, the kernel used may influence the solution to a great extent, making advisable the optimisation of the feature space structure (although the spherical Gaussian kernel has been shown to perform well for several cases); and finally, that there exist some regions of the dataset which should be preferred for over-sampling and that multiple kernel learning techniques should be explored in the future with the purpose of over-sampling.

The authors would also like to stress several lines of future work: Firstly, an analytical methodology for optimising the number of dominant dimensions of the empirical feature space could be developed with the purpose of over-sampling. Secondly, considering a unique methodology combining the techniques proposed in this paper could be accomplished, to analyse how these methods could complement each other. Furthermore, in the context of kernel learning, the over-sampling process could be incorporated in the kernel learning

stage to search the more suitable representation for performing the over-sampling, not only the better class separation. Finally, other intelligent optimisation techniques could be developed for the generation of the synthetic patterns.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[2] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, Oct. 2002.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 281–288, Feb. 2009.

[5] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[6] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 475–482.

[7] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *International Joint Conference on Neural Networks (IJCNN)*, 2008, pp. 1322–1328.

[8] S. Barua, M. M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," in *International Conference on Neural Information Processing (ICONIP)*, 2011, pp. 735–744.

[9] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE - majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. In press, p. 1, 2012.

[10] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012.

[11] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data." in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[12] A. L. N. Fred, T. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds., *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings*, ser. Lecture Notes in Computer Science, vol. 3138. Springer, 2004.

[13] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. Pittsburgh, PA: ACM Press, 1992, pp. 144–152.

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[15] Z.-Q. Zeng and J. Gao, "Improving SVM classification with imbalance data set," in *Proc. of the 16th International Conference on Neural Information Processing: Part I*, ser. ICONIP '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 389–398.

[16] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[17] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, 1999.

[18] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, 2005.

[19] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*. MIT Press, 2002, pp. 367–373.

[20] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.

[21] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *In Proceedings of the 15th European Conference on Machine Learning (ECML)*, 2004, pp. 39–50.

[22] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for SVMs: a case study," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 60–69, Jun. 2004.

[23] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–8.

[24] Y. Liu, A. An, and X. Huang, "Boosting prediction accuracy on imbalanced datasets with SVM ensembles," in *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, ser. PAKDD'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 107–118.

[25] P. Kang and S. Cho, "EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, I. King, J. Wang, L. Chan, and D. Wang, Eds. Springer Berlin Heidelberg, 2006, vol. 4232, ch. 93, pp. 837–846.

[26] X. Hong, S. Chen, and C. J. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 28–41, Jan. 2007.

[27] G. Wu and E. Y. Chang, "KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[28] ——, "Adaptive Feature-Space Conformal Transformation for Imbalanced Data Learning," in *Proceedings of the Twentieth International Conference on Machine Learning*, vol. 20, no. 2, 2003, pp. 816–823.

[29] J. Yuan, J. Li, and B. Zhang, "Learning concepts from large scale imbalanced data sets using support cluster machines," in *Proceedings of the 14th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2006, pp. 441–450.

[30] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21– 27, 1967.

[31] L. W. Johnson and R. D. R. D. Riess, *Numerical analysis*. Reading, Mass. Addison-Wesley Pub. Co. c1982, 1982.

[32] J. T. Y. Kwok and I. W. H. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, Nov. 2004.

[33] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *Journal of Machine Learning Research*, vol. 9, pp. 1875–1908, 2008.

[34] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, Jun. 1998.

[35] M. Ledoux, *The Concentration of Measure Phenomenon*, ser. Mathematical surveys and monographs. American Mathematical Society, 2005.

[36] S. Abe and K. Onishi, "Sparse least squares support vector regressors trained in the reduced empirical feature space," in *Proc. of the 17th international conference on Artificial neural networks*, ser. ICANN. Springer-Verlag, 2007, pp. 527–536.

[37] H. Xiong, "A unified framework for kernelization: The empirical kernel feature space," in *Chinese Conference on Pattern Recognition (CCPR)*, nov. 2009, pp. 1 –5.

[38] M. Ramona, G. Richard, and B. David, "Multiclass feature selection with kernel gram-matrix-based criteria," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 10, pp. 1611–1623, 2012.

[39] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[40] N. Srebro and S. Ben-David, "Learning bounds for support vector machines with learned kernels," in *In Annual Conference On Learning Theory (COLT*. Springer, 2006, pp. 169–183.

[41] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[42] C. Igel and M. Hüsken, "Empirical evaluation of the improved rprop learning algorithms." *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[43] O. Chapelle and A. Rakotomamonjy, "Second order optimization of kernel parameters," in *Neural Information Processing Systems Workshop on Kernel Learning (NIPS)*, 2008.

Over-sampling the minority class in the feature space: Experimental results

This document presents the complete tables of results for the 50 datasets both for the Acc and GM metrics and the 8 methodologies considered in the study. More specifically, TABLE I presents the mean and standard deviation results for Acc and TABLE II the mean and standard deviation results for GM. In the case of TABLE I we also included the results obtained by a majority class rule (MCR) classifier as a baseline result for the problem (i.e., a naïve rule that classify all the patterns as belonging to the majority class).

TABLE I
Acc MEAN AND STANDARD DEVIATION RESULTS (MEAN_{SD}) OBTAINED OVER 30 RUNS BY THE METHODOLOGIES DEVELOPED IN THE PAPER.

Dataset	MCR	SVM	OIS	$OEFS$	$OREFS$	OSK	OGK	$OCPL$	$OPMKL$
ecoli0vs1	$65.00_{1.24}$	*98.56*$_{1.40}$	$98.41_{1.35}$	$98.41_{1.35}$	$98.48_{1.38}$	$\mathbf{98.64_{1.24}}$	$\mathbf{98.64_{2.03}}$	$98.26_{1.42}$	$\mathbf{98.64_{1.41}}$
glass1	$64.49_{1.01}$	$73.52_{3.68}$	*74.40*$_{5.57}$	$74.39_{4.96}$	$73.92_{6.23}$	$\mathbf{77.59_{3.79}}$	$67.16_{6.71}$	$71.26_{5.37}$	$73.83_{5.30}$
wisconsin	$65.01_{0.27}$	$96.83_{0.80}$	$96.78_{0.87}$	$96.83_{0.73}$	$96.85_{0.86}$	$96.83_{0.87}$	$\mathbf{97.07_{0.73}}$	*96.88*$_{0.88}$	$\mathbf{97.07_{0.73}}$
pima	$65.10_{0.23}$	$\mathbf{76.64_{2.57}}$	$74.76_{2.20}$	$74.65_{2.30}$	$74.86_{2.86}$	$74.58_{2.66}$	$74.59_{3.06}$	*74.87*$_{2.83}$	$74.08_{3.71}$
yeast1	$71.09_{0.11}$	*74.32*$_{2.43}$	$69.62_{2.99}$	$70.06_{3.21}$	$70.47_{2.82}$	$71.19_{2.51}$	$71.16_{3.04}$	$71.83_{4.00}$	$\mathbf{75.60_{3.39}}$
haberman	$\mathbf{73.53_{0.53}}$	$64.82_{4.29}$	$67.48_{5.30}$	$67.37_{5.62}$	$67.80_{4.84}$	$69.29_{2.23}$	*71.54*$_{4.85}$	$67.97_{6.00}$	$66.33_{6.87}$
vehicle2	$74.23_{0.30}$	$98.11_{1.14}$	$97.93_{0.98}$	$97.97_{0.93}$	$\mathbf{98.39_{1.14}}$	$93.38_{6.09}$	$97.99_{0.84}$	$97.93_{1.09}$	*98.25*$_{0.95}$
vehicle1	$74.35_{0.29}$	$\mathbf{87.00_{1.05}}$	$83.11_{3.17}$	$84.55_{2.62}$	$84.81_{2.31}$	$81.07_{2.79}$	$81.36_{1.17}$	$85.10_{2.91}$	*85.81*$_{5.05}$
vehicle3	$74.94_{0.29}$	$\mathbf{86.19_{1.54}}$	$81.24_{2.68}$	$82.45_{2.38}$	*82.49*$_{2.25}$	$78.09_{3.82}$	$80.61_{1.35}$	$82.39_{1.80}$	$82.45_{3.34}$
vehicle0	$76.48_{0.26}$	$97.60_{1.18}$	$97.34_{1.11}$	$97.52_{1.22}$	*97.62*$_{1.18}$	$97.58_{1.11}$	$91.37_{5.77}$	$97.40_{1.15}$	$\mathbf{97.70_{1.06}}$
glass0123vs456	$76.17_{1.01}$	$92.84_{2.98}$	$93.54_{3.85}$	$94.01_{3.09}$	*94.71*$_{1.83}$	$91.41_{3.16}$	$91.66_{3.66}$	$\mathbf{94.94_{2.81}}$	$93.43_{3.15}$
ecoli1	$77.09_{0.73}$	$\mathbf{90.83_{4.34}}$	$87.15_{4.51}$	$87.40_{4.94}$	$88.05_{4.61}$	$86.86_{4.59}$	$86.30_{4.90}$	$87.20_{5.94}$	*90.18*$_{7.11}$
newthyroid1	$83.72_{0.00}$	$98.53_{1.14}$	$98.76_{1.70}$	$98.60_{1.79}$	*98.91*$_{1.33}$	$98.76_{1.46}$	$98.53_{1.29}$	$\mathbf{99.07_{1.27}}$	$\mathbf{99.07_{1.27}}$
newthyroid2	$83.72_{0.00}$	$98.37_{1.51}$	$98.53_{1.29}$	*98.84*$_{1.33}$	$98.68_{1.32}$	$98.60_{2.08}$	$\mathbf{99.07_{1.27}}$	$\mathbf{99.07_{1.27}}$	$98.45_{1.54}$
ecoli2	$84.53_{0.76}$	$94.95_{2.27}$	$95.29_{1.90}$	*95.49*$_{1.87}$	$95.00_{2.43}$	$95.29_{1.69}$	$90.82_{4.36}$	$\mathbf{95.54_{2.33}}$	$\mathbf{95.54_{1.47}}$
yeast3	$89.02_{0.17}$	$\mathbf{95.17_{0.73}}$	$92.42_{1.43}$	$92.73_{1.24}$	$92.58_{1.48}$	*92.98*$_{1.68}$	$89.76_{2.96}$	$92.71_{1.65}$	$91.85_{1.48}$
ecoli3	*89.58*$_{0.07}$	$\mathbf{90.97_{1.72}}$	$86.96_{2.82}$	$86.51_{2.02}$	$86.91_{2.76}$	$87.31_{3.86}$	$85.42_{4.76}$	$87.20_{3.89}$	$88.15_{3.40}$
ecoli034vs5	$90.00_{0.00}$	$\mathbf{96.08_{3.98}}$	$94.17_{3.24}$	$94.83_{3.28}$	$93.08_{4.81}$	*95.17*$_{4.04}$	$94.58_{3.09}$	$94.08_{3.31}$	$94.25_{3.48}$
yeast0359vs78	*90.12*$_{0.04}$	$87.05_{4.20}$	$75.79_{4.18}$	$76.45_{4.13}$	$75.59_{5.42}$	$\mathbf{90.48_{2.26}}$	$77.07_{3.62}$	$77.14_{3.52}$	$79.45_{3.83}$
ecoli046vs5	$90.15_{0.13}$	$\mathbf{96.32_{3.24}}$	$94.13_{4.93}$	$94.94_{4.14}$	$91.97_{6.85}$	*95.51*$_{3.68}$	$95.11_{4.55}$	$95.09_{2.98}$	$93.63_{4.89}$
yeast0256vs3789	*90.14*$_{0.20}$	$\mathbf{92.43_{1.45}}$	$88.48_{1.92}$	$89.57_{1.95}$	$89.29_{2.29}$	$89.08_{1.57}$	$89.34_{1.15}$	$89.39_{2.23}$	$88.20_{1.94}$
yeast02579vs368	$90.14_{0.20}$	$\mathbf{96.96_{0.87}}$	$93.47_{2.03}$	$93.59_{1.64}$	$93.92_{1.47}$	*94.62*$_{1.60}$	$93.48_{1.73}$	$93.93_{1.70}$	$92.93_{1.48}$
ecoli0347vs56	$90.27_{0.10}$	$\mathbf{96.50_{2.65}}$	$94.44_{3.10}$	*94.49*$_{3.38}$	$93.58_{2.72}$	$93.27_{3.64}$	$91.46_{3.76}$	$93.65_{2.61}$	$93.26_{3.12}$
ecoli01vs235	$90.17_{0.84}$	$\mathbf{95.55_{1.89}}$	$94.39_{2.47}$	*94.80*$_{1.69}$	$93.99_{4.18}$	$94.33_{2.89}$	$94.67_{1.84}$	$93.78_{2.97}$	$92.62_{4.48}$
yeast2vs4	$90.08_{0.43}$	$\mathbf{94.81_{1.53}}$	$92.70_{1.13}$	$92.96_{1.37}$	$92.31_{2.70}$	$93.00_{1.74}$	$92.61_{1.30}$	*93.25*$_{1.44}$	$91.47_{1.34}$
ecoli067vs35	$90.10_{1.11}$	$\mathbf{96.36_{4.23}}$	$91.60_{4.81}$	$93.12_{5.29}$	$93.50_{4.63}$	$93.26_{5.07}$	*95.97*$_{3.29}$	$85.08_{8.49}$	$89.17_{6.09}$
glass04vs5	$90.23_{2.37}$	$98.54_{2.47}$	$98.91_{2.22}$	*99.07*$_{2.56}$	$98.54_{3.22}$	$97.78_{3.75}$	$\mathbf{100.00_{0.00}}$	$97.78_{4.97}$	$98.95_{2.35}$
ecoli0267vs35	$90.18_{1.18}$	$\mathbf{95.90_{1.88}}$	$90.04_{6.35}$	*94.20*$_{4.47}$	$94.12_{4.32}$	$93.31_{4.43}$	$93.31_{3.50}$	$88.99_{7.23}$	$93.73_{3.74}$
yeast05679vs4	$\mathbf{90.34_{0.40}}$	*89.81*$_{2.62}$	$82.29_{4.57}$	$82.86_{5.11}$	$82.58_{4.97}$	$83.18_{4.45}$	$82.39_{5.79}$	$83.34_{3.79}$	$82.01_{4.33}$
ecoli067vs5	$90.91_{0.00}$	$\mathbf{97.42_{1.55}}$	$92.73_{4.24}$	$91.74_{5.61}$	$93.03_{4.70}$	$92.50_{5.03}$	*93.64*$_{1.90}$	$90.45_{2.96}$	$92.88_{5.33}$
glass016vs2	$\mathbf{91.16_{1.29}}$	$88.39_{3.68}$	$81.17_{5.33}$	$79.19_{4.82}$	$81.62_{4.70}$	*90.21*$_{3.79}$	$83.31_{7.58}$	$82.55_{3.58}$	$79.97_{9.41}$
ecoli01vs5	$91.67_{0.00}$	$\mathbf{96.94_{1.42}}$	$94.31_{3.90}$	$94.93_{3.13}$	$94.17_{4.07}$	$94.24_{3.40}$	$95.42_{4.01}$	$94.58_{4.06}$	*96.25*$_{4.01}$
led7digit02456789vs1	$91.65_{0.58}$	$\mathbf{96.47_{2.29}}$	$92.30_{4.56}$	$74.28_{31.35}$	*92.53*$_{4.17}$	$90.86_{4.94}$	$83.25_{20.39}$	$86.86_{15.60}$	$87.85_{13.03}$
glass06vs5	$91.69_{1.99}$	$\mathbf{100.00_{0.00}}$	$98.79_{3.36}$	$98.00_{4.03}$	$98.79_{2.65}$	$88.12_{10.48}$	$98.14_{2.55}$	$98.43_{4.38}$	*99.70*$_{1.15}$
glass0146vs2	$\mathbf{91.71_{1.34}}$	$87.89_{3.17}$	$81.38_{7.43}$	$80.89_{7.53}$	$82.68_{6.51}$	*89.43*$_{2.89}$	$79.02_{9.38}$	$80.33_{8.97}$	$77.56_{6.07}$
glass2	$\mathbf{92.06_{1.25}}$	$88.69_{3.23}$	$83.62_{4.61}$	$81.84_{5.49}$	$83.02_{6.35}$	*88.76*$_{4.60}$	$86.26_{7.36}$	$82.56_{6.43}$	$84.54_{5.56}$
ecoli0147vs56	$92.47_{0.06}$	$\mathbf{97.55_{1.82}}$	*95.33*$_{2.14}$	$94.02_{2.43}$	$94.72_{2.28}$	$94.62_{2.14}$	$92.46_{2.18}$	$93.63_{2.74}$	$93.57_{5.75}$
cleveland0vs4	$92.50_{1.49}$	*93.32*$_{2.91}$	$91.97_{4.07}$	$91.59_{3.65}$	$90.92_{2.97}$	$92.47_{4.48}$	$\mathbf{94.20_{2.92}}$	$88.42_{3.02}$	$90.77_{3.75}$
ecoli0146vs5	$92.86_{0.00}$	$\mathbf{97.14_{2.37}}$	$95.89_{3.04}$	*96.61*$_{2.79}$	$93.75_{6.34}$	$95.54_{2.08}$	$91.55_{2.89}$	$96.25_{1.84}$	$96.49_{2.02}$
shuttle0vs4	$93.28_{0.14}$	$99.84_{0.14}$	$99.89_{0.14}$	$99.91_{0.13}$	$99.92_{0.13}$	*99.97*$_{0.08}$	$\mathbf{100.00_{0.00}}$	$99.89_{0.14}$	$99.95_{0.12}$
yeast1vs7	*93.46*$_{0.03}$	$\mathbf{94.48_{1.34}}$	$80.05_{5.07}$	$80.63_{4.50}$	$81.14_{4.72}$	$83.28_{6.85}$	$78.87_{5.14}$	$80.53_{4.29}$	$80.67_{6.12}$
ecoli4	$94.05_{0.04}$	$\mathbf{98.22_{0.71}}$	$95.59_{2.19}$	$95.69_{2.48}$	*95.98*$_{2.36}$	$95.15_{3.47}$	$94.64_{4.68}$	$95.04_{1.47}$	$94.65_{2.57}$
pageblocks13vs4	$94.07_{0.56}$	$98.72_{1.44}$	$97.25_{2.27}$	$98.09_{1.88}$	*98.76*$_{1.08}$	$92.44_{3.26}$	$94.48_{2.55}$	$97.03_{2.04}$	$\mathbf{99.36_{0.95}}$
abalone9-18	$94.12_{0.37}$	$\mathbf{95.75_{2.04}}$	$88.10_{2.20}$	$90.65_{2.37}$	$90.83_{1.48}$	$92.61_{3.83}$	$90.15_{1.58}$	$90.84_{1.90}$	*95.49*$_{1.03}$
glass016vs5	$95.12_{1.18}$	$\mathbf{97.57_{2.28}}$	$96.58_{2.74}$	$97.03_{2.87}$	*97.30*$_{2.75}$	$96.48_{2.15}$	$86.49_{21.87}$	$96.22_{4.10}$	$\mathbf{97.57_{2.39}}$
yeast2vs8	$95.85_{0.02}$	$\mathbf{97.92_{0.67}}$	$96.71_{1.80}$	$96.40_{1.91}$	$85.40_{9.24}$	$96.47_{1.19}$	$96.88_{2.45}$	$97.44_{1.25}$	*97.65*$_{0.99}$
shuttle2vs4	$95.35_{1.70}$	$\mathbf{99.23_{1.56}}$	$96.66_{7.87}$	$97.56_{3.42}$	*97.82*$_{3.14}$	$97.79_{4.07}$	$90.38_{13.60}$	$97.29_{3.06}$	$97.29_{3.06}$
yeast4	$\mathbf{96.56_{0.15}}$	*95.28*$_{1.28}$	$86.82_{1.75}$	$86.69_{1.63}$	$87.17_{1.96}$	$85.38_{0.99}$	$86.12_{1.63}$	$86.50_{1.09}$	$84.42_{4.04}$
yeast5	*97.04*$_{0.15}$	$\mathbf{97.99_{0.70}}$	$94.62_{1.17}$	$95.01_{1.22}$	$95.09_{1.23}$	$93.94_{1.14}$	$95.23_{1.49}$	$95.55_{1.10}$	$95.45_{1.61}$
yeast6	$\mathbf{97.64_{0.00}}$	*97.36*$_{0.83}$	$92.62_{1.18}$	$92.81_{1.40}$	$92.96_{1.48}$	$94.62_{2.54}$	$89.04_{1.87}$	$92.08_{1.60}$	$91.58_{1.39}$

TABLE II
GM MEAN AND STANDARD DEVIATION RESULTS (MEAN_{SD}) OBTAINED OVER 30 RUNS BY THE METHODOLOGIES DEVELOPED IN THE PAPER.

Dataset	SVM	OIS	$OEFS$	$OREFS$	OSK	OGK	$OCPL$	$OPMKL$
ecoli0vs1	$98.11_{2.03}$	$98.06_{1.70}$	$98.00_{1.85}$	$98.11_{1.86}$	*$98.28_{1.70}$*	$\mathbf{98.31_{2.39}}$	$97.77_{2.01}$	$98.17_{2.06}$
glass1	$66.76_{4.97}$	$69.53_{6.93}$	$69.90_{5.28}$	*$70.20_{7.03}$*	$70.13_{6.91}$	$67.64_{7.05}$	$68.44_{5.15}$	$\mathbf{70.82_{6.22}}$
wisconsin	$96.72_{0.79}$	$96.81_{0.93}$	$96.85_{0.82}$	*$96.95_{0.91}$*	$96.85_{0.98}$	$\mathbf{96.97_{0.80}}$	$96.85_{1.08}$	$\mathbf{96.97_{0.62}}$
pima	$69.72_{4.07}$	$74.02_{2.61}$	$74.10_{2.47}$	$74.11_{3.29}$	$74.28_{3.15}$	$\mathbf{75.17_{3.42}}$	*$74.40_{2.80}$*	$73.50_{2.74}$
yeast1	$60.47_{3.48}$	$70.77_{2.64}$	$71.10_{2.95}$	$71.34_{2.39}$	$71.23_{2.11}$	$72.24_{3.66}$	*$72.34_{3.73}$*	$\mathbf{72.64_{4.14}}$
haberman	$33.91_{12.55}$	$61.47_{4.47}$	$61.09_{5.61}$	$\mathbf{62.96_{5.91}}$	$61.16_{3.38}$	$61.31_{8.33}$	$62.36_{5.75}$	*$62.38_{5.08}$*
vehicle2	$97.51_{1.43}$	$97.29_{1.31}$	$97.42_{1.16}$	$\mathbf{97.90_{1.58}}$	$85.83_{14.67}$	*$97.57_{1.17}$*	$97.29_{1.46}$	$97.48_{1.36}$
vehicle1	$82.86_{2.54}$	$83.79_{3.22}$	$86.00_{2.60}$	*$86.07_{2.28}$*	$81.46_{2.79}$	$83.06_{1.84}$	$\mathbf{86.14_{3.35}}$	$84.43_{7.27}$
vehicle3	$79.88_{2.91}$	$83.23_{2.89}$	*$83.85_{2.71}$*	$83.59_{3.18}$	$80.52_{3.12}$	$82.40_{3.14}$	$\mathbf{83.88_{2.81}}$	$81.36_{3.59}$
vehicle0	$96.76_{1.33}$	$97.61_{1.10}$	$97.66_{1.08}$	$\mathbf{97.79_{1.01}}$	$97.06_{1.07}$	$93.13_{3.90}$	*$97.75_{0.92}$*	$96.91_{1.17}$
glass0123vs456	$88.08_{7.41}$	$92.85_{4.86}$	$93.26_{3.77}$	$\mathbf{94.23_{2.56}}$	$87.82_{5.92}$	$89.81_{5.46}$	*$94.09_{3.54}$*	$92.03_{3.52}$
ecoli1	$86.64_{5.85}$	$87.55_{4.67}$	$88.06_{4.29}$	$87.86_{5.42}$	$87.63_{5.11}$	$88.08_{4.28}$	$\mathbf{89.06_{5.56}}$	*$88.80_{6.02}$*
newthyroid1	$97.91_{2.78}$	$99.25_{1.03}$	$99.16_{1.09}$	*$99.35_{0.80}$*	$99.25_{0.88}$	$97.91_{2.90}$	$\mathbf{99.44_{0.77}}$	$\mathbf{99.44_{0.77}}$
newthyroid2	$96.02_{3.80}$	$99.11_{0.78}$	$99.10_{1.46}$	$98.60_{2.18}$	*$99.16_{1.26}$*	$\mathbf{99.44_{0.77}}$	$98.24_{3.22}$	$96.27_{3.81}$
ecoli2	$90.56_{3.44}$	$93.84_{3.84}$	*$94.09_{3.85}$*	$93.67_{4.02}$	$93.56_{3.53}$	$91.59_{4.49}$	$\mathbf{94.25_{4.18}}$	$93.41_{3.14}$
yeast3	$85.07_{2.91}$	$91.70_{1.96}$	*$92.40_{1.91}$*	$91.63_{2.48}$	$91.14_{2.52}$	$90.97_{2.16}$	$91.57_{2.47}$	$\mathbf{92.44_{2.20}}$
ecoli3	$74.62_{9.16}$	$86.72_{5.66}$	$86.98_{5.60}$	$86.55_{4.87}$	$86.92_{4.58}$	$\mathbf{88.79_{3.94}}$	$87.27_{5.22}$	*$87.28_{5.91}$*
ecoli034vs5	$83.74_{16.20}$	$89.17_{11.34}$	$88.38_{11.35}$	$87.78_{11.11}$	$88.58_{11.71}$	$\mathbf{89.45_{11.72}}$	$89.14_{11.41}$	*$89.23_{11.49}$*
yeast0359vs78	$48.98_{16.56}$	$74.00_{4.55}$	$73.67_{4.84}$	$72.88_{7.00}$	$40.21_{17.39}$	$\mathbf{75.86_{6.32}}$	*$74.01_{5.58}$*	$72.27_{7.75}$
ecoli046vs5	$82.90_{17.98}$	$89.12_{11.38}$	$89.20_{11.53}$	$87.16_{12.37}$	*$89.55_{11.79}$*	$\mathbf{89.69_{12.81}}$	$\mathbf{89.69_{12.65}}$	$86.76_{14.10}$
yeast0256vs3789	$70.20_{4.49}$	$78.81_{5.47}$	$79.93_{5.15}$	$79.88_{5.29}$	$79.64_{5.47}$	*$80.15_{5.06}$*	$80.02_{5.26}$	$\mathbf{80.17_{4.90}}$
yeast02579vs368	$88.50_{4.73}$	$89.59_{3.40}$	$90.06_{3.33}$	$89.60_{3.99}$	$89.70_{4.43}$	$\mathbf{90.33_{3.88}}$	*$90.20_{3.92}$*	$90.09_{3.64}$
ecoli0347vs56	$87.05_{13.42}$	$88.45_{13.99}$	$89.04_{12.48}$	$90.21_{8.07}$	$90.44_{8.95}$	$\mathbf{91.27_{9.80}}$	*$90.56_{8.19}$*	$87.57_{12.60}$
ecoli01vs235	$78.79_{20.56}$	$84.28_{15.35}$	$84.40_{18.07}$	$83.62_{17.77}$	$87.66_{15.39}$	$\mathbf{89.84_{11.98}}$	$88.43_{10.52}$	*$88.71_{11.83}$*
yeast2vs4	$80.51_{6.42}$	$86.21_{4.62}$	$86.74_{4.68}$	$87.01_{5.32}$	$86.86_{5.82}$	$\mathbf{89.56_{4.87}}$	$87.97_{4.44}$	*$88.06_{2.74}$*
ecoli067vs35	$76.57_{39.62}$	$77.11_{31.43}$	$79.36_{31.74}$	$80.27_{28.84}$	$80.20_{31.49}$	$\mathbf{85.85_{23.45}}$	$79.71_{20.50}$	*$82.20_{21.87}$*
glass04vs5	$91.59_{20.50}$	$89.70_{30.42}$	*$99.48_{1.46}$*	$99.17_{1.84}$	$98.74_{2.15}$	$\mathbf{100.00_{0.00}}$	$98.71_{2.89}$	$99.40_{1.34}$
ecoli0267vs35	$78.70_{15.88}$	$79.87_{13.73}$	$86.23_{11.09}$	$83.91_{11.06}$	$86.20_{11.16}$	*$86.38_{12.18}$*	$81.23_{13.18}$	$\mathbf{86.49_{11.71}}$
yeast05679vs4	$61.62_{12.41}$	$76.06_{7.78}$	$76.29_{8.26}$	$75.33_{8.68}$	$76.21_{8.76}$	$\mathbf{78.10_{7.64}}$	$76.40_{9.27}$	*$77.59_{6.77}$*
ecoli067vs5	*$86.44_{8.87}$*	$86.41_{6.28}$	$85.52_{6.60}$	$85.69_{6.96}$	$85.43_{7.15}$	$\mathbf{89.37_{6.86}}$	$85.41_{6.15}$	$86.09_{6.44}$
glass016vs2	$43.50_{28.65}$	$74.27_{11.75}$	$71.86_{12.25}$	*$75.78_{11.82}$*	$31.11_{26.05}$	$74.85_{11.52}$	$\mathbf{81.28_{10.17}}$	$73.13_{12.32}$
ecoli01vs5	$85.57_{12.50}$	$86.71_{10.47}$	$89.01_{8.15}$	$88.48_{8.75}$	$86.70_{10.52}$	$87.35_{12.24}$	*$89.83_{7.45}$*	$\mathbf{90.73_{8.09}}$
led7digit02456789vs1	$\mathbf{89.84_{8.01}}$	*$88.24_{5.91}$*	$72.57_{32.26}$	$88.12_{6.50}$	$69.33_{35.91}$	$84.52_{12.60}$	$85.46_{17.22}$	$86.67_{9.37}$
glass06vs5	$\mathbf{100.00_{0.00}}$	$99.31_{1.91}$	$97.08_{7.53}$	$99.32_{1.49}$	$69.61_{36.69}$	$98.99_{1.39}$	$99.10_{2.54}$	*$99.83_{0.64}$*
glass0146vs2	$32.03_{33.98}$	$65.39_{28.08}$	$71.47_{17.32}$	*$73.39_{18.04}$*	$31.05_{25.93}$	$\mathbf{73.55_{14.28}}$	$72.94_{17.10}$	$73.03_{15.02}$
glass2	$28.77_{32.41}$	$75.99_{13.19}$	$78.81_{14.39}$	$78.56_{12.61}$	$29.58_{28.34}$	$\mathbf{85.56_{10.57}}$	$79.33_{14.05}$	*$79.74_{16.77}$*
ecoli0147vs56	$87.85_{9.35}$	$90.07_{5.15}$	$89.94_{5.49}$	$\mathbf{91.03_{3.99}}$	*$90.94_{4.99}$*	$90.17_{4.45}$	$90.07_{5.90}$	$88.21_{6.92}$
cleveland0vs4	$59.81_{38.10}$	$79.12_{29.82}$	$81.48_{29.26}$	$84.63_{16.37}$	$88.15_{17.68}$	$89.13_{18.11}$	*$90.29_{6.45}$*	$\mathbf{91.64_{7.34}}$
ecoli0146vs5	$81.12_{19.29}$	$88.70_{13.30}$	*$89.72_{11.77}$*	$88.59_{10.46}$	$\mathbf{89.73_{11.36}}$	$87.56_{10.26}$	$87.11_{14.07}$	$87.55_{14.70}$
shuttle0vs4	$99.54_{0.79}$	$99.94_{0.07}$	$99.95_{0.07}$	$99.96_{0.07}$	*$99.99_{0.04}$*	$\mathbf{100.00_{0.00}}$	$99.94_{0.07}$	$99.97_{0.07}$
yeast1vs7	$48.29_{23.18}$	$74.62_{6.32}$	$76.21_{5.05}$	$74.89_{6.08}$	$61.02_{31.48}$	$74.51_{5.26}$	*$76.31_{5.73}$*	$\mathbf{76.39_{5.74}}$
ecoli4	$89.40_{5.45}$	$91.87_{6.75}$	$91.92_{6.78}$	$92.08_{6.90}$	*$92.81_{5.88}$*	$92.23_{7.99}$	*$92.81_{6.17}$*	$\mathbf{93.85_{6.13}}$
pageblocks13vs4	$93.89_{6.76}$	$97.39_{3.32}$	$98.11_{3.05}$	*$98.41_{3.08}$*	$84.75_{18.44}$	$97.02_{1.39}$	*$98.41_{1.10}$*	$\mathbf{99.66_{0.51}}$
abalone9vs18	$63.77_{17.37}$	$88.95_{3.63}$	$89.90_{3.33}$	$\mathbf{91.50_{2.55}}$	$52.44_{44.27}$	$89.29_{0.85}$	$90.49_{4.90}$	*$90.51_{5.49}$*
glass016vs5	$76.54_{39.94}$	$83.36_{34.01}$	*$91.00_{25.30}$*	$88.84_{30.15}$	$66.18_{39.10}$	$88.08_{21.91}$	$\mathbf{92.39_{12.32}}$	$79.28_{40.33}$
yeast2vs8	*$72.83_{13.62}$*	$72.28_{13.13}$	$68.45_{21.87}$	$\mathbf{75.86_{12.41}}$	$72.20_{14.47}$	$72.35_{14.47}$	$71.01_{18.53}$	$72.71_{13.53}$
shuttle2vs4	$94.14_{11.92}$	$92.69_{12.10}$	$93.26_{11.60}$	$94.30_{10.85}$	$\mathbf{98.82_{2.19}}$	*$94.49_{7.80}$*	$93.11_{11.50}$	$94.02_{10.72}$
yeast4	$52.40_{10.54}$	$80.34_{8.23}$	$82.48_{2.87}$	$79.60_{8.52}$	*$82.92_{0.46}$*	$\mathbf{84.08_{4.22}}$	$81.86_{3.54}$	$82.82_{1.11}$
yeast5	$83.93_{4.65}$	$96.80_{1.37}$	$96.44_{3.06}$	$96.69_{3.01}$	$96.82_{0.61}$	*$97.51_{0.78}$*	$96.57_{3.65}$	$\mathbf{97.62_{0.85}}$
yeast6	$63.38_{17.84}$	$87.77_{6.04}$	$88.10_{6.32}$	$87.94_{6.09}$	$86.73_{7.40}$	$87.15_{6.71}$	*$88.21_{6.67}$*	$\mathbf{88.42_{7.57}}$

5.2. Graph-based approaches for over-sampling in the context of ordinal regression

As stated before, the imbalanced nature of ordinal problems is a current challenge for researchers. For most ordinal datasets, there are some classes that are naturally much more probable than others (specially when the number of classes is high). However, to the best of our knowledge, the case of imbalanced ordinal classification problems has not been tackled yet, despite its relevance for real world applications. Nonetheless, there are some methods that have been shown to work well in general for several ordinal and imbalanced metrics.

Although standard over-sampling methods [23] could also be applied to ordinal regression, it is clear that the new synthetic samples will be obtained ignoring the ordering of the labelling space, and this can result in classifiers more prone to commit errors involving several categories in the ordinal scale. Motivated by some preliminary studies and by this issue, the following paper proposes different graph-based over-sampling approaches for imbalanced ordinal classification problems. The proposals are based on the concept of neighbourhood graphs and extends the graph construction strategy with the aim of capturing the underlying latent manifold which reflects the implicit ordering among the classes. This is the first time this graph view of the over-sampling process is given, and it is very convenient for including the necessary ordering constraints in the ordinal regression context. In this sense, we develop three over-sampling approaches (where the ordinal information is included using different graph-based strategies) and we support our hypotheses with an extensive experimental analysis over 30 ordinal and imbalanced datasets. We also propose a cost-sensitive extension of the ordinal SVM classifier, which gives more importance to minority class errors, and compare the results obtained.

The results show that the inclusion of ordinal information in the over-sampling process improves both the classification and the ordering of minority classes, thus being a suitable approach for datasets with an imbalanced and ordinal structure. On the other hand, the experiments also shows that a cost-sensitive approach may in general improve the base performance, but still without reaching the results of over-sampling methods.

Graph-Based Approaches for Over-sampling in the context of Ordinal Regression

M. Pérez-Ortiz, P.A. Gutiérrez, *Member, IEEE,* C. Hervás-Martínez, *Member, IEEE* and X. Yao, *Fellow, IEEE*

Abstract—The classification of patterns into naturally ordered labels is referred to as ordinal regression or ordinal classification. Usually, this classification setting is by nature highly imbalanced, because there are classes in the problem that are a priori more probable than others. Although standard over-sampling methods can improve the classification of minority classes in ordinal classification, they tend to introduce severe errors in terms of the ordinal label scale, given that they do not take the ordering into account. A specific ordinal over-sampling method is developed in this paper for the first time in order to improve the performance of machine learning classifiers. The method proposed includes ordinal information by approaching over-sampling from a graph-based perspective. The results presented in this paper show the good synergy of a popular ordinal regression method (a reformulation of support vector machines) with the graph-based proposed algorithms, and the possibility of improving both the classification and the ordering of minority classes. A cost-sensitive version of the ordinal regression method is also introduced and compared with the over-sampling proposals, showing in general lower performance for minority classes.

Index Terms—Over-sampling, imbalanced classification, ordinal regression, ordinal classification

1 Introduction

Ordinal regression (also known as ordinal classification) can be defined as a relatively new learning paradigm whose aim is to learn a prediction rule for ordered categories. This paradigm shares properties of classification and regression. In contrast to multinomial classification, there exists some ordering among the elements of $\mathcal{Y}$ (the labelling space) and both standard classifiers and the zero-one loss function do not capture and reflect this ordering. Concerning regression, $\mathcal{Y}$ is a non-metric space (thus distances among categories are unknown) and a finite set.

Ordinal classification problems arise in several areas such as economy [1], medicine [2], [3] or image ranking [4], to name a few. For an explanatory example, consider the case of financial trading where an agent intends to predict not only whether to buy an asset, but also the amount of investment. The different situations could be categorised as {"no investment", "little investment", "big investment", "huge investment"}. In this case, the natural order among the classes can be appreciated, as well as the necessity of penalising differently the misclassification errors (it should not be considered equal misclassifying a "no investment" instance with a "huge investment" one than misclassifying it with "little investment"). Other explanatory examples could be the use of a Likert scale for rating the quality of certain article/service or when trying to predict different levels of an illness. For all these cases, there are some classes that are naturally much more probable than others (specially when the number of classes is high) and therefore the problem present an imbalanced character. Following the previous example, it is reasonable to expect a lower number of "huge investment" situations than the number of "little investment" ones.

In particular, the classification paradigm when one or several categories present a much lower prior probability is known as imbalanced classification [5], [6] and it generally poses a serious hindrance for the learning process of machine learning algorithms. Furthermore, it has been shown that there are others factors such as the existence of noisy and non-representative samples or the size of the dataset that could be involved in the nature of the class imbalance problem [5], [6]. In this sense, different approaches have been developed over the last decades in the context of binary classification [7], [8], multinomial classification [9], [10], [11] and even regression [12]. Different perspectives have been considered: sampling data approaches (over-sampling groups of interesting and rare examples or under-sampling majority classes) and algorithmic approaches (e.g. cost-sensitive learning). However, to the best of our knowledge, the case of imbalanced ordinal classification problems has not been tackled yet, despite its relevance for real world applications. Nonetheless, there are some methods that have been shown to work well in general for several ordinal and imbalanced metrics, such as the ensemble approach in [13] where various order hypotheses were formulated

- *M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez are with the Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, C2 building, 14004 - Córdoba, Spain, e-mail: {i82perom,pagutierrez,chervas}@uco.es.*

- *Xin Yao is with the Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA), School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K., e-mail: X.Yao@cs.bham.ac.uk.*

and fused (although the method was not specifically designed for the imbalanced setting and no experiments were performed in highly imbalanced datasets).

Concerning previous studies, the cost-sensitive learning setting has been proved to lead to over-fitting [14] in some cases, thus data approaches are usually preferred. In the same vein, some studies suggest that over-sampling is more useful and powerful than under-sampling for highly imbalanced and complex datasets [6], [15], given the potential loss of meaningful information of under-sampling techniques.

Although standard over-sampling methods could also be applied to ordinal regression, the new synthetic samples will be obtained without taking into account the ordering of the labelling space and this can result in classifiers more prone to commit errors of several categories in the ordinal scale (this will be shown in the experimental part of this paper). Motivated by the studies previously analysed and by this empirical conclusion, this paper proposes different graph-based over-sampling approaches for imbalanced ordinal classification problems. The proposed methods are used in conjunction with the well-known SMOTE algorithm [7] and a popular reformulation of the support vector machine paradigm (SVM) for ordinal classification [16]. This classifier has been chosen because it is one of the most successful, well-known and widely used in this context, despite the fact that the usual formulation of the soft-margin maximisation paradigm is focused on improving overall performance, consequently harming the classification of minority classes. In this sense, we develop three over-sampling approaches (where the ordinal information is included using different graph-based strategies) and we support our hypotheses with an extensive experimental analysis over 30 ordinal and imbalanced datasets. We also propose a cost-sensitive extension of the ordinal SVM classifier, which gives more importance to minority class errors, and compare the results obtained. The inclusion of ordinal information in the over-sampling process is shown to improve both the classification and the ordering of minority classes, thus being a suitable approach for datasets with an imbalanced and ordinal structure.

The paper is organized as follows: Section II introduces some useful notions for the paper; Section III formally presents the different proposed methods; Section IV exposes the experimental study and analyses the results; and finally, Section V outlines some conclusions and future work.

2 Related techniques

Consider a training sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$ generated i.i.d. from a (unknown) joint distribution $P(\mathbf{x}, y)$, where $\mathcal{X} \subseteq \mathbb{R}^K$ and $\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_Q\}$. In the ordinal regression setup, the labelling space is ordered due to the data ranking structure ($\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_Q$, where $\prec$ denotes this order information). Let N be the number of patterns in the training sample, N_q the number of samples for the q-th class and X_q the set of patterns belonging to class $\mathcal{C}_q$. In this section, we first describe the SMOTE algorithm and then the support vector method for ordinal regression, as this will be the base technique used for the proposed resampling methodologies. Finally, we also propose a cost-sensitive version of support vector ordinal regression based on the use of different costs for each class.

2.1 SMOTE technique and extension to regression

One of the most widely used techniques for over-sampling is the SMOTE algorithm [7]. The process is very simple: the method consists on generating new instances in the line that connects one randomly chosen point and one of its k-nearest neighbours [17], both belonging to the minority class. As will be discussed in the experimental section, the application of the standard SMOTE algorithm tends to improve minority class classification at the cost of decreasing ordering performance metrics associated to these classes.

A very recent study [12] has shown the usefulness of over-sampling rare but interesting examples in the context of regression. The crucial points in this sense are how to decide rare examples and the computation of the target variable for the new synthetic patterns (recall that in the case of classification the new pattern corresponds to the class that is to be over-sampled). Rare examples are discovered using the target variable prior probability density function and new examples are generated in the same manner than for the SMOTE algorithm (by linear interpolation of nearest rare patterns). The target variable for a synthetic point is created using a weighted average of the target variables of the two seed examples and the weights are calculated as an inverse function of the distance to the two seed examples. For the regression case, there exist an observable target space, but, in the ordinal regression setting, although sometimes a latent space generating the ordered labels is assumed, it is always unobservable. Consequently, labels for new synthetic samples could not be derived in a principled way, precluding the application of regression SMOTE for ordinal classification problems. However, this regression-based study could be considered as a motivation for one of the ideas developed in this paper (the use of a probability function for the constructed intra-class edges).

2.2 Support Vector Ordinal Regression with Implicit Constraints (SVORIM)

The SVM paradigm [18] is considered the most common kernel learning method for statistical pattern recognition. In this sense, some works in the literature have been focused on the reformulation of this successful paradigm to tackle ordinal regression problems [16], [19], [20]. In the context of binary SVM, the so-called slack-variables ξ_i are used to replace hard margins with soft margins

[21]. However, as stated before, the usual SVM formulation of the soft-margin maximisation is focused on improving overall performance and does not take into account the class imbalance.

One of the most widely used SVM-based methods for ordinal regression is presented in the work of Chu and Keerthi [16]. The idea is to seek for $Q-1$ parallel discriminant hyperplanes (or a projection vector $\mathbf{w}$) and the scalar bias $b_1 \leq \ldots \leq b_{Q-1}$ in order to properly separate the data into ordered classes by modelling ranks as intervals on the real line (i.e., trying to capture the underlying but unobservable latent variable). One of the two alternatives discussed in [16] is known as Support Vector Ordinal Regression with Implicit Constraints (SVORIM). For SVORIM, the whole training sample (considering all classes) is used for the determination of each threshold and samples in all the categories are allowed to contribute errors for each hyperplane. This can be a crucial handicap for imbalanced classification problems, because errors from patterns very far in the ordinal scale still have an important impact on minority classes, and therefore the hindrance posed for learning these classes can be even bigger.

More specifically, the learning problem for the SVORIM algorithm is defined as follows:

$$\min_{\mathbf{w},\mathbf{b},\xi,\xi^*} \frac{1}{2}||\mathbf{w}|| + C\sum_{q=1}^{Q-1}\left(\sum_{j=1}^{q}\sum_{i=1}^{N_q}\xi_{ji}^{q} + \sum_{j=q+1}^{Q}\sum_{i=1}^{N_q}\xi_{ji}^{*q}\right), \quad (1)$$

subject to the constraints:

$$\mathbf{w}\cdot\Phi(\mathbf{x}_i^j) - b_k \leq -1 + \xi_{ji}^{q},\ \ \xi_{ji}^{q} \geq 0 \quad (2)$$
$$\text{for } j = \{1,\ldots,q\},\ i = \{1,\ldots,N_q\}$$
$$\mathbf{w}\cdot\Phi(\mathbf{x}_i^j) - b_{k+1} \geq +1 - \xi_{ji}^{*q},\ \ \xi_{ji}^{*q} \geq 0 \quad (3)$$
$$\text{for } j = \{q+1,\ldots,Q\},\ i = \{1,\ldots,N_q\}$$

where $\mathbf{b} \in \mathbb{R}^{Q-1}$, and ξ_{ji}^{q} and ξ_{ji}^{*q} are the slacks for the q-th parallel hyperplane (defined for the left and right part of the hyperplanes, respectively).

2.3 Cost-sensitive SVORIM (CS-SVORIM)

One of the possibilities for considering the imbalanced nature of the datasets tackled in this paper is to use different cost parameters for each class (i.e. by assigning a higher cost for minority classes, which is similar to modify the loss function [22]). In this section, we propose to apply this idea to SVORIM, in order to compare the results obtained by the proposed over-sampling methods to a cost-sensitive approach. The SVORIM learning problem is extended as follows:

$$\min_{\mathbf{w},\mathbf{b},\xi,\xi^*} \frac{1}{2}||\mathbf{w}|| + \sum_{q=1}^{Q-1}\left(\sum_{j=1}^{q}\sum_{i=1}^{N_q} C_j\cdot\xi_{ji}^{q} + \sum_{j=q+1}^{Q}\sum_{i=1}^{N_q} C_j\cdot\xi_{ji}^{*q}\right),$$

where different costs $C_j, j = 1,\ldots,Q-1$, have been included for the slacks of the different classes. These costs are adjusted based on the imbalanced ratio of the classes, in such a way that minority classes receive more attention than majority ones. Given a value for the hyperparameter C, this can be done by setting $C_j = C\cdot IR_j$, where IR_j is the imbalance ratio associated to $\mathcal{C}_j$ (being higher as the number of patterns of the corresponding classes are lower). The exact details of the computation of IR_j for ordinal imbalanced datasets are explained in Section 4.1.

3 Graph-based methodologies for ordinal over-sampling

The main idea of the three methods proposed in this paper is to generate new patterns in a space where there exists an ordering relation between the patterns. All these methods are based on analysing the data from a graph-based perspective, in order to easily include the ordering information in the synthetic pattern generation process.

3.1 Construction of a representative graph between adjacent classes

It is important to note that the notion of graph has been used indirectly with the purpose of over-sampling in the machine learning literature (e.g. for the SMOTE technique) since it is usually assumed for practical applications that a graph is a closed estimation for the real manifold structure of the data [23]. As said, these methods create new synthetic patterns in the line that connects a pattern and one of its k-nearest neighbours (both belonging to the minority class). In this paper, we exploit the fact that this process is equivalent to constructing a graph based on the neighbourhood information of the minority class patterns and creating synthetic points in the edges of the graph. In our case, we use the graph notion to include further information about the order of the classes. The idea is to obtain a better representation of the classes based on a neighbourhood analysis and taking their ordering relationship into account. Roughly speaking, we are aiming at the construction of a representative graph G that connects nearest points not only belonging to the same class but also to close classes in the ranking scale. The idea of analysing the input space of ordinal regression problems using a distance relation but with the aim of reconstructing and estimating the underlying latent variable has been previously studied in [24].

Consider a graph of n vertices, $G = (V, E)$, where V corresponds to the vertices of the graph and $E \subseteq [V]^2$ to the edges. In our case, some of the patterns in the training dataset form the set of vertices, $V = \{v_1, v_2, \ldots, v_n\} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, and the different edges connect pairs of patterns:

$$E = \{e_{i,j}\} = \{(v_i, v_j)\} = \{(\mathbf{x}_i, \mathbf{x}_j)\}, 1 \leq i \leq n, 1 \leq j \leq n. \quad (4)$$

Let q be the index of the class we want to over-sample. For our purpose, we construct a graph G_q for class $\mathcal{C}_q$

based on three subgraphs $G_{q-1,q}$, $G_{q,q}$ and $G_{q,q+1}$ (each of these based on a neighbourhood analysis) in such a way that $G_q = G_{q-1,q} \cup G_{q,q} \cup G_{q,1+1}$. More formally:

- $G_{q-1,q} = (V_{q-1,q}, E_{q-1,q})$, where the edges $E_{q-1,q}$ of the graph, are the intersection of two sets, which are found by analysing the neighbourhood based on a distance relation d, i.e.:

$$E_{q-1,q} = \mathcal{N}_d(X_{q-1}, X_q, k) \cap \mathcal{N}_d(X_q, X_{q-1}, k). \quad (5)$$

For two arbitrary sets X_1 and X_2, the k-neighbourhood of X_1 with respect to X_2, $\mathcal{N}_d(X_1, X_2, k)$, is defined in the following way:

$$\mathcal{N}_d(X_1, X_2, k) = \{e_{i,j} \mid \quad (\mathbf{x}_i \in X_1) \wedge (\mathbf{x}_j \in X_2) \wedge (\mathbf{x}_j \in nn_d(\mathbf{x}_i, X_2, k))\}, \quad (6)$$

where $nn_d(\mathbf{x}_i, X_2, k)$ is the set of vertices from X_2 which are the k-nearest neighbours of $\mathbf{x}_i$ based on a distance relation d. In this way, $\mathcal{N}_d(X_1, X_2, k)$ represents the set of edges connecting X_1 to their k-nearest neighbours in X_2. We considered only the intersection of both edge sets because this gives us only the edges which are present in both directions, thus strictly connecting regions on the border of the two classes. In this sense, the value of k will determine the broadness of the region of the class frontier that is considered. The vertices of the graph, $V_{q-1,q}$, are those points which can be found in the resulting edge set, $E_{q-1,q}$:

$$V_{q-1,q} = \{\mathbf{x}_i \mid \exists \mathbf{x}_j, \ (e_{i,j} \in E_{q-1,q})\}. \quad (7)$$

- $G_{q,q} = (V_{q,q}, E_{q,q})$, $V_{q,q} = X_q$ and $E_{q,q} = \mathcal{N}_d(X_q, X_q, k)$.
- $G_{q,q+1} = (V_{q,q+1}, E_{q,q+1})$ is constructed using the analogue procedure used for $G_{q-1,q}$ but for classes $\mathcal{C}_q$ and $\mathcal{C}_{q+1}$.

Note that $G_{q-1,q}$ or $G_{q,q+1}$ can be the empty set $\varnothing$ if $\mathcal{C}_q$ is an extreme class in the ordinal scale. An example of the resulting graph is given in the left part of Fig. 1, where this process has been applied to classes $\mathcal{C}_1$ and $\mathcal{C}_3$. Recall that for the nominal version of the SMOTE algorithm, only the subgraph $G_{q,q}$ is taken into account for the generation of new synthetic patterns for $\mathcal{C}_q$.

With this information, we can construct an adjacency matrix $\mathbf{A}_q = (a_{ij})_{n \times n}$ (being n the order of the graph) in such a way that:

$$a_{ij} := \begin{cases} 1 & \text{if } e_{i,j} \in E_q, v_i \in V_q, v_j \in V_q, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

where $e_{i,j}$ is the edge incident to both v_i and v_j.

3.2 Over-sampling of points in the edges of the constructed graph

The creation of a new synthetic point $\mathbf{s}_p$ is made by selecting one edge $e_{i,j} \in E_q$. The way the graph has been created makes one of the extremes of the edge be a point of X_q, while the other extreme can be or not included in X_q (see Fig. 1). Let $\mathbf{x}_i$ be the extreme included in X_q, $\mathbf{x}_i \in X_q$. We interpolate both extremes $\mathbf{x}_i$ and $\mathbf{x}_j$ (which both belong to the set of vertices V_q from G_q) as follows:

$$\mathbf{s}_p = \mathbf{x}_i + \delta \cdot (\mathbf{x}_j - \mathbf{x}_i), \quad (9)$$

where δ is a random number generated from a given distribution. If $\mathbf{x}_j \in X_q$ the distribution is set to be the uniform one $U[0, 1]$ (as for the nominal SMOTE); otherwise a different distribution is chosen so that the new synthetic pattern is more prone to fall near the $\mathbf{x}_i$ example. We allow points to be created between a point of the minority class and one point of the adjacent classes, which is a differentiating characteristic with respect to standard SMOTE. We hypothesise that, in an ordinal regression setting, this way of generating new patterns can help to respect better the ordinal structure of the dataset. Furthermore, the creation of points in the "intra-class" region could also help to define better the boundary of the minority class (note that this boundary may not be well-defined because of the low information available for this class). It could be useful in cases where the number of minority patterns is very low (for example, 2 or 3 patterns). In this case, the directions used to create new synthetic points that the minority SMOTE technique consider are very limited. However, with our approach there will be more possible directions for the creation of synthetic points, which could, in general, avoid over-fitting. This method will be named in the experiments as ordinal graph-based over-sampling via neighbourhood information using a probability function for the intra-class edges (OGO-NI).

3.3 Identification of the shortest paths in the graph for over-sampling

One of the main hypothesis in ordinal regression is that the distance to adjacent classes is lower than the distance to non-adjacent classes. Therefore, it can be said that ideally there exists a latent distance-based manifold of the output variable that results in $\mathcal{C}_q$ lying in the space between $\mathcal{C}_{q-1}$ and $\mathcal{C}_{q+1}$. The method proposed in this subsection is based on the assumption that by over-sampling the patterns lying in these "intra-class" area the ordinal information in the dataset could be improved. This idea could be useful as well to detect outliers in the minority class, which should not be used for over-sampling purposes.

The main idea is to use the graph information obtained in the previous step to detect the patterns from $\mathcal{C}_q$ that are spatially located in the underlying manifold between $\mathcal{C}_{q-1}$ and $\mathcal{C}_{q+1}$. To do so, we use the graph notion of shortest path.

In graph theory, the shortest path problem is the problem of finding a path between two vertices in a graph such that the sum of the weights of its constituent edges is minimised. To do so, first take into account that the constructed graph is undirected. Therefore, the notion of path is defined as a sequence of vertices

$p_{1z} = (v_1, v_2, \ldots, v_z) \in V_q^z$ such that v_i is adjacent to v_{i+1} for $1 \leq i < z$ (and therefore $e_{i,i+1}$ exists).

Moreover, given a real-valued weight function $f : E_q \rightarrow \mathbb{R}$ that assigns a cost to each edge and an undirected graph G_q, the shortest path from v to v' is the path $p_{1z} = (v_1, \ldots, v_z)$ (where $v_1 = v$ and $v_z = v'$) that over all possible z minimises the sum $\sum_{i=1}^{z-1} f(e_{i,i+1})$, where $e_{i,i+1} \in G_q$.

In our case, we select the Euclidean distance as the weight function, (the one used for the neighbourhood analysis):

$$f(e_{i,j}) = ||\mathbf{x}_i - \mathbf{x}_j||_2, \tag{10}$$

where $||\cdot||_2$ is the $L2$-norm operator.

To identify the patterns in $\mathcal{C}_q$ lying in the latent manifold between $\mathcal{C}_{q-1}$ and $\mathcal{C}_{q+1}$, we compute the shortest paths from all the vertices in $V_{q-1,q}$ to all the vertices in $V_{q,q+1}$ by using the well known Dijkstra's algorithm [25]. Denote by P_q the set of paths obtained from this process, where $p_{i,j} \in P_q$ is the shortest path between $v_i \in V_{q-1,q}$ and $v_j \in V_{q,q+1}$. Note that this method is not possible when considering the over-sampling of an extreme class in the ordinal scale. In those cases, we consider the shortest paths between all the vertices in $V_{q-1,q}$ or $V_{q,q+1}$ (depending on the ranking of the extreme class) and $V_{q,q}$.

Up to this point, we could consider two different approaches for over-sampling $\mathcal{C}_q$:

- First, we can create synthetic points only in the edges of the graph G_q connecting minority class patterns (edge $e_{i,j} \in E_{q,q}$ for patterns $\mathbf{x}_i \in X_q$ and $\mathbf{x}_j \in X_q$) and such that $e_{i,j}$ is at least contained in one of the paths of P_q. This method will be named in the experiments as ordinal graph-based over-sampling via interior shortest paths (OGO-ISP).
- Second, as stated in subsection 3.2, we also could exploit the intra-class information by creating patterns in edges that connect patterns from $V_{q-1,q}$ to $V_{q,q}$, or from $V_{q,q}$ to $V_{q,q+1}$, using a probability weighting function. That is, allowing $\mathbf{x}_j$ to be included in X_{q-1} or X_{q+1}, but also constructing $e_{i,j}$ to be at least contained in one of the paths of P_q. This method will be named in the experiments as ordinal graph-based over-sampling via shortest paths using a probability function for the intra- class edges (OGO-SP).

Fig. 1 represents the differences when including the shortest paths information in the graph construction step. Minority classes in the dataset are $\mathcal{C}_1$ and $\mathcal{C}_3$. It can be seen that, in the right part of the figure, some edges are removed because they do not belong to any of the shortest paths. In this way, the shortest paths construction (the right plot) helps to detect the outliers and to reinforce those parts of the minority class which respect more the ordering structure.

4 Experimental results

The proposed methodologies have been tested considering Support Vector Ordinal Regression with Implicit Constraints (SVORIM) [16] and the well-known SMOTE algorithm [7]. 30 ordinal benchmark datasets from the UCI repository with different imbalance ratios (proportion of majority patterns with respect to minority ones) have been used for the analysis to test the performance of the methods in different situations. Some of the original datasets have been relabelled to obtain different imbalanced distributions (see Table 1, where eucalyptus123vs4vs5 stands for the eucalyptus dataset, $\mathcal{C}_1$ groups the original labels 1, 2 and 3, $\mathcal{C}_2$ is the original label 4 and $\mathcal{C}_3$ is the original label 5). The datasets from the UCI are specific ordinal regression datasets (although they have been commonly tackled as nominal classification). However, some of the ordinal regression benchmark datasets (wisconsin, stock, housing, machine, triazines, auto and abalone) provided by Chu et. al [26] were considered since they are widely used in the ordinal regression literature [27], [16]. These datasets do not originally represent ordinal classification tasks but regression ones. To turn regression into ordinal classification, the target variable is discretised into Q different bins (representing classes, in this case Q was assigned to 5 or 10), with equal sizes of the bins, which usually results in a high imbalance.

4.1 Datasets

The characteristics for all the datasets used for the experiments can be seen in Table 1. From this Table, one could appreciate that, in the ordinal regression setting, extreme classes are the ones more prone to present an imbalanced distribution, given that they usually represent rare events. The mean imbalance ratio (IR) included in this table is defined as

$$IR = \frac{1}{Q}\sum_{q=1}^{Q} IR_q, \tag{11}$$

where IR_q corresponds to the imbalance ratio associated to $\mathcal{C}_q$:

$$IR_q = \frac{\sum_{j \neq q} N_j}{Q \cdot N_q}. \tag{12}$$

Several works in the literature have considered the over-sampling of classes that present a IR value (for the considered class) higher than 1.5 [28], [9]. In this work we consider the same threshold for deciding to apply over-sampling to a certain class and for computing the number of necessary synthetic patterns for class $\mathcal{C}_q$. That is, we compute the patterns needed for IR_q^* to be lower than 1.5[1]. However, in a multiclass setting, IR_q depends on the patterns belonging to the other classes, so when some patterns are added to a certain class, the IR^* values for the rest of classes change. Therefore, we use an iterative procedure, where in each iteration, the number of patterns needed for obtaining a IR^* lower than 1.5 for the minority class is obtained, and the method is run

1. IR_q^* represents the modified IR_q value when the number of patterns N_q is modified.

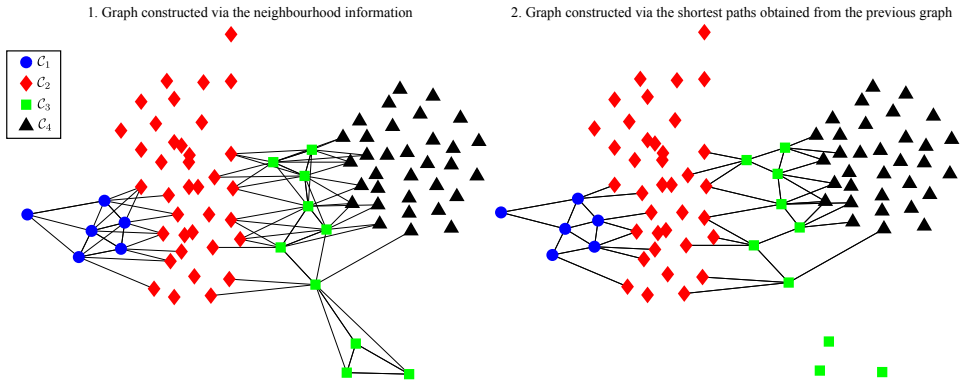


Fig. 1. Comparison of two different methodologies for the construction of a graph for a two-dimensional ordinal synthetic dataset. The graph in the left part represents the neighbourhood information between patterns (method described in subsection 3.1). The graph in the right part shows the graph obtained when considering only the edges that are included in the set of shortest paths (method described in subsection 3.3).

until all classes have $IR_q^* \leq 1.5,\ q = 1,\ ldots, Q$. The same procedure is considered for all the over-sampling techniques. The classes which are to be over-sampled for all the datasets are highlighted in bold face in Table 1.

4.2 Metrics for the evaluation of the results

Several measures can be considered for evaluating ordinal classifiers. The most common ones in machine learning are the mean absolute error (MAE) and the mean zero-one error (MZE) [29], being $MZE = 1 - Acc$, where Acc is the accuracy or correct classification rate. MAE is the average deviation (in number of categories) between the predicted label and the target label [30]. However, these measures may not be the best option when the costs of different errors vary markedly (as in ordinal classification problems) or when the dataset is unbalanced (as in this case). Because of that, this work makes use of other measures to evaluate an ordinal classifier performance. First, we consider the maximum mean absolute error ($MMAE$) [31], which is the MAE value considering only the patterns from the class with the greatest distance between true labels and predicted ones:

$$MMAE = \max\{MAE_q; q \in \{1, \ldots, Q\}\}, \qquad (13)$$

where MAE_q is the MAE value considering only the patterns from the q-th class:

$$MAE_q = \frac{1}{N_q} \sum_{i=1}^{N_q} |\mathcal{O}(y_i) - \mathcal{O}(\hat{y}_i)|, \qquad (14)$$

where $\mathcal{O}(\cdot)$ represents the ranking of the class and $\hat{y}_i$ is the predicted label for $\mathbf{x}_i$. $MMAE$ values range from 0 to $Q-1$. This measure was recently proposed [31] and its advantage is its represents the individual performance for the worst ordered class, in such a way that a low $MMAE$ represents a low error for all classes of the problem (including minority ones). We considered this measure due to its specific nature for ordinal and imbalanced classification problems.

To make clear how the imbalance nature of some datasets can affect the performance of the minority classes, Fig. 2 shows the cross-validation error during model selection for the two SVORIM hyper-parameters (the cost C and the kernel width α). It can be seen that the metrics MAE and $MMAE$ present a very different nature for highly imbalanced datasets (such as machine5 and abalone5). As MAE is a global measure which gives the same importance to all the patterns, models with a very low MAE value can be hiding high MAE values for the minority classes, so that the classifier assigns labels for these classes very far away from their real one. This results in the fact that MAE and $MMAE$ optimum are obtained at different parameters combination, as opposed to slightly imbalanced datasets (toy and eucalyptus123vs4vs5) where the optimum region is similar. This motivates the use of the $MMAE$ measure for this study and shows that these two measures are conflicting objectives for highly skewed data distributions when optimising the SVORIM parameters.

The geometric mean of the sensitivities for each class (GMS) is also taken into account in this study, although it is a measure designed for nominal (imbalanced) classification problems and it does not take the ordinal nature of the problem into account. This measure is defined as the geometric mean of the correct classification rates for all classes:

$$GMS = \sqrt[Q]{\prod_{q=1}^{Q} S_q}, \qquad (15)$$

where $S_q = \frac{1}{N_q} \sum_{i=1}^{N_q} (I(\mathcal{O}(\hat{y}_i) = \mathcal{O}(y_i)))$ is the sensitivity of the q-th class, i.e. the percentage of patterns correctly predicted as belonging to the q-th class with respect to the total number of examples in this class.

It is important to recall that in the ordinal regression paradigm, accuracy and sensitivity are no longer a

TABLE 1
Characteristics of the datasets. M corresponds to the total number of patterns, K to the dimensionality of the input space and Q to the number of classes in the problem.

Dataset	M	K	Q	Pattern distr.	IR per class	Mean IR
toy	300	2	5	(35,87,79,68,31)	(**1.53**,0.49,0.56,0.68,**1.68**)	0.99
eucalyptus123vs4vs5	736	91	3	(417,214,105)	(0.25,0.82,**2.00**)	1.02
wisconsin5	194	32	5	(67,41,43,24,19)	(0.38,0.74,0.71,1.41,**1.87**)	1.02
eucalyptus1vs2vs345	736	91	3	(180,107,449)	(1.03,**1.94**,0.21)	1.06
wisconsin10	194	32	10	(46,21,28,13,25,18,14,10,9,10)	(0.31,0.81,0.59,1.35,0.71,1.02,1.35,**1.71,1.97,1.97**)	1.18
stock	950	9	10	(48,110,108,119,168,104,104,103,64,22)	(**1.88**,0.77,0.78,0.70,0.47,0.81,0.81,0.81,1.38,**4.35**)	1.28
newthyroid	215	5	3	(30,150,35)	(**2.00**,0.15,**1.73**)	1.29
bondrate	57	37	4	(6,33,12,5)	(**1.85**,0.19,0.92,**2.38**)	1.33
housing5	506	13	5	(77,239,123,36,31)	(1.11,0.22,0.62,**2.61,3.10**)	1.53
automobile12vs345vs6	205	71	3	(25,153,27)	(**2.50**,0.11,**2.10**)	1.57
balance-scale	625	4	3	(288,49,288)	(0.39,**4.00**,0.39)	1.59
heating	768	8	8	(20,265,112,51,119,85,82,34)	(**4.67**,0.24,0.73,**1.77**,0.68,1.00,1.06,**2.64**)	1.6
ERA	1000	4	9	(92,142,181,172,158,118,88,31,18)	(1.10,0.68,0.50,0.53,0.60,0.83,1.15,**3.51,5.84**)	1.64
auto	392	7	5	(91,131,101,59,10)	(0.65,0.40,0.58,1.14,**7.15**)	1.98
triazines5	186	60	5	(7,10,26,86,57)	(**5.36**,**3.27**,1.26,0.23,0.46)	2.12
LEV	1000	4	5	(93,280,403,197,27)	(**1.94**,0.51,0.30,0.81,**7.30**)	2.17
housing10	506	13	10	(22,55,85,154,84,39,29,7,10,21)	(**2.27**,0.80,0.49,0.23,0.50,1.21,**1.62,7.48,4.64,2.43**)	2.17
SWD	1000	10	4	(32,352,399,217)	(**7.56**,0.46,0.38,0.90)	2.33
automobile	205	71	6	(3,22,67,54,32,27)	(**12.58**,1.43,0.33,0.47,0.90,1.11)	2.8
ESL12vs3vs456vs7vs89	488	4	5	(14,38,351,62,23)	(**6.45**,**2.41**,0.08,1.39,**3.87**)	2.84
machine5	209	6	5	(152,27,13,7,10)	(0.07,1.36,**2.92**,**6.04**,**4.26**)	2.93
triazines10	186	60	10	(4,3,2,8,11,15,36,50,45,12)	(**4.53**,**4.53**,**13.80**,**2.22**,**1.64**,1.16,0.41,0.28,0.31,1.44)	3.03
machine10	209	6	10	(115,37,21,6,8,5,3,4,4,6)	(0.08,0.46,0.94,**3.02**,**2.50**,**3.80**,**7.70**,**5.10**,**5.10**,**3.80**)	3.25
car	1728	21	4	(1210,384,69,65)	(0.11,0.88,**5.98**,**6.36**)	3.33
ERA1vs23456vs7vs8vs9	1000	4	5	(92,771,88,31,18)	(**1.97**,0.06,**2.07**,**6.32**,**10.51**)	4.19
ESL	488	4	9	(2,12,38,100,116,135,62,19,4)	(**20.22**,**4.41**,1.29,0.43,0.36,0.29,0.77,**2.79**,**13.44**)	4.89
winequality-red	1599	11	6	(10,53,681,638,199,18)	(**24.81**,**4.96**,0.23,0.25,1.17,**15.21**)	7.77
winequality-white	4898	11	7	(20,163,1457,2198,880,175,5)	(**34.84**,**4.16**,0.34,0.18,0.65,**3.86**,**131.04**)	25.01
abalone5	4177	10	5	(448,3036,557,129,7)	(**1.66**,0.08,1.30,**6.26**,**125.08**)	26.88
abalone10	4177	10	10	(17,431,1648,1388,432,125,100,29,4,3)	(**23.99**,0.87,0.15,0.20,0.87,**3.23**,**4.08**,**14.81**,**104.30**,**156.50**)	30.9

The classes that are over-sampled are in bold face in the IR per class column.

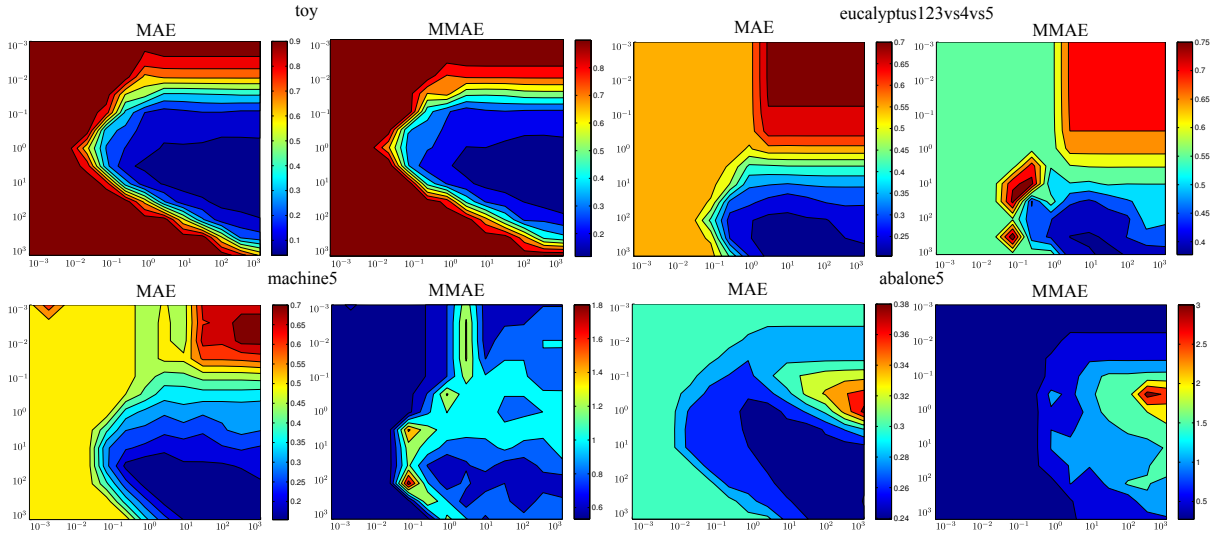


Fig. 2. Cross-validation error using a 5-fold procedure for the different combinations of parameters associated to the SVORIM method. The abscissa axis represents for all the cases the cost parameter C and the ordinate axis the kernel parameter α.

proper measure to take into account, because they lack a penalization for misclassifying patterns in classes very far away from the real class (in the ordinal scale). In this way, GMS better reflects errors committed for the minority class, but it does not consider what kind of error has been committed. $MMAE$ measure does not have this limitation.

4.3 Methods compared

The methods compared were run and optimised under the same conditions and using the same model selection process. Regarding the experimental setup, a holdout stratified technique was applied to divide the datasets 30 times, using 75% of the patterns for training and the remaining 25% for testing. The partitions were the same for all methods and one model was obtained and

evaluated (in the test set), for each split. Finally, the results are taken as the mean and standard deviation of the measures over the 30 test sets.

The parameters of each algorithm are chosen using a nested cross-validation considering only the training set (specifically, a 5-fold method). The cross-validation criteria (the measure used to select the best parameter combination) is the $MMAE$ for all the methods, because it is the metric that we will consider for the evaluation of the results. The kernel selected for all the algorithms is the Gaussian one, $K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)$ where σ is the kernel width. The kernel width was selected within the values $\{10^{-2}, 10^0, 10^2\}$, as well as the cost parameter associated with SVM methods. $k = 5$ nearest neighbours are used for all the algorithms.

For all SMOTE-based methods, the iterative process described in Section 4.1 is used to decide the classes to be over-sampled and the number of patterns to generate for each class. For an extensive analysis, several methods are compared:

- SVORIM without over-sampling the minority classes (SVORIM).
- SVORIM with multiclass nominal over-sampling of the minority classes (MSMOTE) [7]. Nominal over-sampling consists of simply applying SMOTE to the selected classes, considering only the class to over-sample when generating synthetic patterns.
- SVORIM with costs according to the imbalance ratio of each class as explained in Subsection 2.3 (CS-SVORIM).
- SVORIM with ordinal graph-based over-sampling via neighbourhood information using a probability function for the intra-class edges (OGO-NI) (method described in Subsection 3.2).
- SVORIM with ordinal graph-based over-sampling via interior shortest paths (OGO-ISP) (first method described in Subsection 3.3).
- SVORIM with ordinal graph-based over-sampling via shortest paths using a probability function for the intra-class edges (OGO-SP) (second method described in Subsection 3.3).

As said, when creating new synthetic patterns by convex combination of two patterns belonging to the minority class, $\delta \sim U[0,1]$, i.e. any point of the line connecting both extremes can be the new synthetic pattern with equal probability. However, when one of the extremes belongs to an adjacent class, one should avoid to generate new patterns too close to the adjacent class. In this way, we consider the random variable δ in Eq. (9) to be gamma-distributed , i.e., $\delta \sim \Gamma(a,b)$. The parameters have been fixed to the following values: $a = 2$ and $b = 0.15$ (the Gamma density distribution obtained can be seen in Fig. 3). The choice of these parameters is not arbitrary. Note that δ is a random variable that influences the position of the new synthetic point. In this sense, a zero value ($\delta = 0$) means that the synthetic pattern will be the extreme of the minority class (and $\delta = 1$ means that the synthetic pattern will be the extreme of the adjacent class). Fig. 3 shows how the new synthetic patterns are more prone to fall near the minority class patterns (but never too close which will result in replicating these patterns, which has been shown to lead to over-fitting [14]).

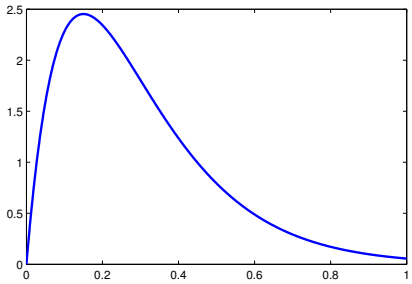


Fig. 3. Gamma density distribution for $a = 2$ and $b = 0.15$.

The source codes in Matlab for all the over-sampling methods developed in this paper are available, together with all of the datasets and partitions on the website associated with this paper[2].

4.4 Analysis of the results obtained

First of all, a graphical representation of the behaviour of the over-sampling methodologies for the housing10 dataset is given in Fig. 4. This figure includes a representation of the projection done by SVORIM algorithm, which projects patterns to a real line (linear in the feature space but nonlinear in the original space) and divides the classes by a set of thresholds. This is done by the vast majority of ordinal regression methods [29], because it is a way to uncover the latent variable which is the origin of the ordinal labels. The estimated latent variables for the SVORIM and OGO-NI methods and the housing10 dataset are shown in this figure. Dashed lines represent the thresholds which divide the different classes. The plot in the top represents the SVORIM algorithm and the plot in the bottom the OGO-NI method. It can be seen that given the imbalanced nature of the dataset and the SVORIM methodology, some of the classes are almost obviated in the latent space (e.g. $\mathcal{C}_8$, where the two thresholds are so close that a new pattern will be hardly classified in this class). Therefore, SVORIM without over-sampling tends to misclassify minority class patterns at a expense of an overall error minimisation. On the contrary, in the bottom plot, one can observe that the ordinal synthetic over-sampling technique helps to fix fairer thresholds for the minority classes. Furthermore, it can be seen that the new synthetic patterns maintain the ordinal information of the dataset.

Tables 2, 3 and 4 shows respectively the MAE, GMS and $MMAE$ results in mean and standard deviation for the 6 methods and the 30 datasets considered. From this Table, it can be seen that the SVORIM method

2. http://www.uco.es/grupos/ayrna/GBOforOR

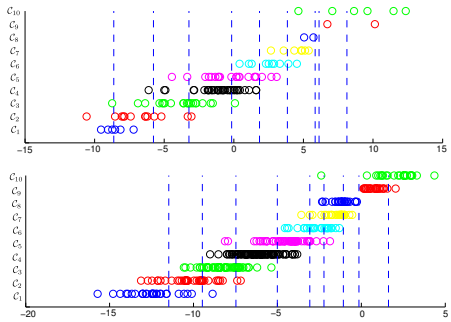


Fig. 4. Training estimated latent variable for the housing10 dataset. The plot in the top represents the original SVORIM algorithm and the plot in the bottom the SVORIM model after applying OGO-NI. The abscissa axis represents the real class of each sample and the ordinate axis the projected values for the patterns (i.e., the estimated latent variable). Vertical dashed lines are thresholds separating the different classes.

obtained the best results in MAE for most of the datasets (apparently this improvement in MAE seems more obvious for datasets with a medium or high imbalance ratio). However, when considering the rest of metrics, it can be seen that this good performance in MAE is accompanied by very poor results for GMS and $MMAE$. This low performance for minority classes is, in general, not acceptable for imbalanced problems. This is very representative of the fact that SVORIM minimises the overall error at the expense of the minority classes. For the case of SVORIM, this result is consistent with previous works stating that it performs better in terms of absolute deviations in number of classes [16].

In order to better summarise these results, Table 5 shows the test mean rankings in terms of Acc, MAE, GMS and $MMAE$, for all the methods considered in these experiments along all the 30 datasets. For each dataset, a ranking of 1 is given to the best method, in average, and a 6 is given to the worst one. We included the Acc measure to show the relative correlation between the ranking results obtained for this metric and the MAE measure for most of the methods (because of this reason we do not consider the Acc measure for the statistical tests). From this Table, it can be seen that most of the over-sampling methods perform similarly for GMS improving to a great extent the results of SVORIM (although OGO-SP performs slightly better than the rest). This is also applicable for $MMAE$. It can also be appreciated that the use of the costs derived from the imbalanced ratio per class helps to improve the results for GMS and $MMAE$ but deteriorates at the same time the results for Acc and MAE compared to the results of SVORIM. As opposed to this, the MSMOTE technique obtains acceptable performance for all the metrics except $MMAE$ (which is not rare since this method is not specially designed for ordinal classification).

To quantify whether a statistical difference exists among the algorithms compared, a procedure is employed to compare multiple classifiers in multiple datasets [32]. Table 5 also shows the result of applying the non-parametric statistical Friedman's test (for a significance level of $\alpha = 0.05$) to the mean Acc, MAE, GMS and $MMAE$ rankings. It can be seen that the test rejects the null-hypothesis that all of the algorithms perform similarly in mean ranking for all the metrics (note that for $MMAE$ the differences are larger).

On the basis of this rejection and following the guidelines in [32], we consider the best performing methods in MAE and $MMAE$ (i.e., SVORIM and OGO-SP) as control methods for the following tests. Furthermore, we also consider the method OGO-ISP as a control method, because it obtained a promising balance between both metrics, and the CS-SVORIM technique, in order to analyse the potential differences between cost-sensitive and over-sampling approaches. The GMS metric was not included for the analysis, because all the over-sampling algorithms obtained significant differences when compared to $SVORIM$ but there were no significant differences between them (although OGO-SP obtained the highest ranking). We compare these three methods to the rest according to their rankings. It has been noted that the approach of comparing all classifiers to each other in a post-hoc test is not as sensitive as the approach of comparing all classifiers to a given classifier (a control method). One approach to this latter type of comparison is the Holm's test. The test statistics for comparing the i-th and j-th method using this procedure is:

$$z = \frac{R_i - R_j}{\sqrt{\frac{J(J+1)}{6T}}},$$

where J is the number of algorithms, T is the number of datasets and R_i is the mean ranking of the i-th method. The z value is used to find the corresponding probability from the table of the normal distribution, which is then compared with an appropriate level of significance α. Holm's test adjusts the value for α in order to compensate for multiple comparisons. This is done in a step-up procedure that sequentially tests the hypotheses ordered by their significance. We will denote the ordered p-values by $p_1, p_2, \ldots, p_q$ so that $p_1 \leq p_2 \leq \ldots \leq p_q$. Holm's test compares each p_i with $\alpha^*_{\text{Holm}} = \alpha/(J-i)$, starting from the most significant p value. If p_1 is below $\alpha/(J-1)$, the corresponding hypothesis is rejected and we allow to compare p_2 with $\alpha/(J-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on.

From Table 6, several conclusions can be drawn. First, it can be seen that SVORIM significantly outperforms most of the over-sampling algorithms for MAE (except for OGO-ISP and OGO-SP where non-significant differences were found). This result is obvious if we take into account that the MAE measure does not consider the imbalanced nature of the problem. On the contrary, when considering $MMAE$ the SVORIM algorithm obtains significantly worse results than all the developed ordi-

TABLE 2
MAE mean and standard deviations (Mean $\pm$ SD) obtained over 30 runs by all the methodologies compared.

MAE	SVORIM	MSMOTE	CS-SVORIM	OGO-NI	OGO-ISP	OGO-SP
toy	*0.052 ± 0.026*	0.056 ± 0.032	0.081 ± 0.041	0.058 ± 0.032	**0.051 ± 0.025**	0.053 ± 0.028
eucalyptus123vs4vs5	0.259 ± 0.031	*0.258 ± 0.033*	0.259 ± 0.032	0.259 ± 0.030	*0.258 ± 0.031*	**0.256 ± 0.032**
wisconsin5	1.250 ± 0.087	**1.213 ± 0.094**	1.253 ± 0.083	1.228 ± 0.091	*1.220 ± 0.099*	1.221 ± 0.102
eucalyptus1vs2vs345	*0.185 ± 0.024*	0.189 ± 0.021	0.271 ± 0.031	**0.184 ± 0.023**	*0.185 ± 0.024*	*0.185 ± 0.020*
wisconsin10	2.476 ± 0.171	2.430 ± 0.147	2.576 ± 0.307	**2.407 ± 0.150**	2.420 ± 0.227	*2.418 ± 0.217*
stock	0.219 ± 0.033	0.232 ± 0.042	0.250 ± 0.046	0.218 ± 0.030	*0.217 ± 0.030*	**0.214 ± 0.025**
newthyroid	0.028 ± 0.019	0.028 ± 0.021	0.071 ± 0.043	0.029 ± 0.023	**0.025 ± 0.021**	*0.027* ± 0.025
bondrate	*0.584 ± 0.105*	0.600 ± 0.112	0.667 ± 0.181	0.598 ± 0.119	0.589 ± 0.109	**0.576 ± 0.128**
housing5	**0.249 ± 0.031**	0.252 ± 0.031	0.356 ± 0.037	*0.250 ± 0.031*	*0.250 ± 0.032*	0.251 ± 0.030
automobile12vs345vs6	0.102 ± 0.040	0.103 ± 0.038	0.197 ± 0.056	**0.097 ± 0.037**	*0.100 ± 0.041*	0.101 ± 0.034
balance-scale	*0.049 ± 0.016*	0.072 ± 0.017	0.085 ± 0.020	0.060 ± 0.020	**0.048 ± 0.014**	0.061 ± 0.018
heating	0.317 ± 0.024	0.305 ± 0.022	0.355 ± 0.027	0.307 ± 0.024	**0.304 ± 0.023**	*0.305 ± 0.027*
ERA	*1.231 ± 0.060*	1.249 ± 0.047	**1.226 ± 0.058**	1.244 ± 0.046	1.242 ± 0.055	1.245 ± 0.045
auto	**0.275 ± 0.058**	0.295 ± 0.044	0.342 ± 0.053	0.295 ± 0.044	*0.291 ± 0.040*	0.296 ± 0.049
triazines5	**0.709 ± 0.074**	0.730 ± 0.090	1.077 ± 0.117	*0.724 ± 0.070*	0.738 ± 0.089	0.726 ± 0.089
LEV	**0.410 ± 0.036**	*0.431 ± 0.032*	0.585 ± 0.046	0.433 ± 0.027	0.435 ± 0.036	0.441 ± 0.031
housing10	**0.488 ± 0.065**	0.505 ± 0.067	0.608 ± 0.077	0.498 ± 0.068	*0.491 ± 0.053*	0.496 ± 0.057
SWD	**0.447 ± 0.030**	0.482 ± 0.039	0.572 ± 0.039	*0.476 ± 0.038*	0.480 ± 0.037	0.481 ± 0.032
automobile	0.371 ± 0.078	*0.367 ± 0.074*	0.463 ± 0.111	0.369 ± 0.073	0.369 ± 0.075	**0.366 ± 0.072**
ESL12vs3vs456vs7vs89	**0.166 ± 0.032**	0.182 ± 0.034	0.426 ± 0.043	0.185 ± 0.037	*0.181 ± 0.031*	0.190 ± 0.033
machine5	0.185 ± 0.041	0.191 ± 0.048	0.386 ± 0.114	**0.182 ± 0.063**	*0.184 ± 0.044*	**0.182 ± 0.053**
triazines10	**1.333 ± 0.129**	1.502 ± 0.139	2.003 ± 0.129	1.777 ± 0.273	*1.491 ± 0.085*	1.674 ± 0.282
machine10	**0.477 ± 0.090**	0.529 ± 0.135	0.833 ± 0.128	0.498 ± 0.083	0.496 ± 0.086	*0.484 ± 0.083*
car	*0.013 ± 0.005*	0.014 ± 0.005	0.057 ± 0.024	0.017 ± 0.006	**0.012 ± 0.005**	0.016 ± 0.005
ERA1vs23456vs7vs8vs9	**0.269 ± 0.018**	*0.278 ± 0.021*	0.906 ± 0.062	0.284 ± 0.030	0.291 ± 0.036	0.290 ± 0.031
ESL	*0.302 ± 0.040*	0.343 ± 0.043	**0.297 ± 0.042**	0.327 ± 0.043	0.333 ± 0.047	0.338 ± 0.041
winequality-red	*0.424 ± 0.017*	0.515 ± 0.027	**0.423 ± 0.016**	0.518 ± 0.025	0.520 ± 0.022	*0.515 ± 0.024*
winequality-white	**0.498 ± 0.027**	0.588 ± 0.019	0.832 ± 0.161	0.587 ± 0.013	*0.583 ± 0.015*	0.586 ± 0.014
abalone5	**0.247 ± 0.021**	0.270 ± 0.013	0.727 ± 0.044	0.273 ± 0.015	*0.269 ± 0.012*	0.276 ± 0.013
abalone10	**0.517 ± 0.030**	*0.629 ± 0.031*	1.370 ± 0.405	0.633 ± 0.027	0.631 ± 0.034	0.644 ± 0.031

The best performing method is in **bold** face and the second one in *italics*.

TABLE 3
GMS mean and standard deviations (Mean $\pm$ SD) obtained over 30 runs by all the methodologies compared.

GMS	SVORIM	MSMOTE	CS-SVORIM	OGO-NI	OGO-ISP	OGO-SP
toy	*94.62 ± 2.56*	94.14 ± 3.43	92.16 ± 4.50	93.87 ± 3.47	**94.66 ± 2.52**	94.31 ± 2.95
eucalyptus123vs4vs5	65.97 ± 5.56	*66.34 ± 5.20*	65.97 ± 5.55	66.30 ± 5.26	*66.34 ± 5.24*	**66.86 ± 5.38**
wisconsin5	**2.87 ± 8.95**	1.16 ± 6.36	2.18 ± 8.31	*2.74 ± 8.64*	0.00 ± 0.00	2.36 ± 8.99
eucalyptus1vs2vs345	72.41 ± 3.46	72.56 ± 3.33	71.11 ± 4.43	*73.23 ± 3.40*	72.52 ± 3.63	**73.58 ± 3.47**
wisconsin10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
stock	77.23 ± 4.20	76.17 ± 4.27	75.55 ± 4.13	77.38 ± 3.32	*77.51 ± 3.48*	**77.72 ± 3.06**
newthyroid	94.81 ± 4.16	94.90 ± 4.08	95.36 ± 4.02	94.95 ± 4.48	*95.50 ± 4.33*	**95.72 ± 4.40**
bondrate	0.00 ± 0.00	0.00 ± 0.00	**1.92 ± 10.54**	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
housing5	72.60 ± 4.16	73.09 ± 3.75	73.08 ± 3.23	73.16 ± 3.67	*73.26 ± 3.88*	**73.32 ± 3.70**
automobile12vs345vs6	78.35 ± 9.45	78.79 ± 8.57	79.42 ± 7.99	**80.83 ± 8.61**	78.78 ± 9.63	*80.09 ± 9.22*
balance-scale	*93.65 ± 2.56*	93.30 ± 2.48	92.31 ± 2.80	93.21 ± 2.74	**93.76 ± 2.52**	93.10 ± 2.94
heating	54.40 ± 11.53	*68.39 ± 3.52*	**69.58 ± 2.98**	67.36 ± 4.18	67.89 ± 3.94	67.76 ± 3.84
ERA	0.00 ± 0.00	**2.66 ± 6.92**	0.00 ± 0.00	0.65 ± 3.56	0.70 ± 3.86	*0.72 ± 3.94*
auto	11.07 ± 25.25	*50.37 ± 31.60*	**55.84 ± 26.17**	47.28 ± 32.25	41.39 ± 34.92	49.64 ± 31.29
triazines5	0.00 ± 0.00	2.70 ± 10.27	0.98 ± 5.35	1.26 ± 6.91	*2.83 ± 10.77*	**4.24 ± 13.22**
LEV	4.50 ± 13.75	37.03 ± 21.34	**46.64 ± 16.62**	38.91 ± 20.03	38.34 ± 19.81	*42.82 ± 17.54*
housing10	0.00 ± 0.00	16.74 ± 26.08	15.79 ± 26.73	*17.69 ± 27.59*	**19.03 ± 27.44**	16.71 ± 26.04
SWD	10.64 ± 18.05	*47.46 ± 4.64*	**54.19 ± 3.59**	46.56 ± 5.93	47.18 ± 5.37	46.08 ± 10.40
automobile	61.00 ± 24.93	58.57 ± 27.14	53.17 ± 28.26	*63.24 ± 22.17*	58.31 ± 27.05	**63.34 ± 22.21**
ESL12vs3vs456vs7vs89	58.74 ± 8.25	66.39 ± 8.33	*66.63 ± 6.25*	**66.73 ± 8.68**	64.71 ± 8.17	66.18 ± 8.00
machine5	22.81 ± 31.15	24.23 ± 31.31	25.87 ± 30.82	**37.32 ± 33.91**	32.12 ± 33.15	*36.99 ± 33.43*
triazines10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
machine10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
car	94.72 ± 2.27	*95.81 ± 2.23*	95.53 ± 2.23	95.53 ± 2.02	**96.28 ± 1.88**	95.63 ± 1.80
ERA1vs23456vs7vs8vs9	0.00 ± 0.00	*21.10 ± 19.40*	**42.01 ± 12.96**	10.36 ± 16.49	9.15 ± 16.20	2.23 ± 8.89
ESL	11.01 ± 25.20	22.90 ± 28.72	8.86 ± 23.15	**37.52 ± 29.23**	31.58 ± 30.19	*33.27 ± 29.76*
winequality-red	0.00 ± 0.00	**9.50 ± 16.17**	0.00 ± 0.00	5.58 ± 12.80	*9.30 ± 15.77*	6.68 ± 13.69
winequality-white	0.00 ± 0.00	2.94 ± 9.08	**6.51 ± 14.87**	5.03 ± 11.47	*5.87 ± 12.03*	5.01 ± 11.46
abalone5	0.00 ± 0.00	28.77 ± 24.00	25.60 ± 24.54	*29.10 ± 24.25*	**29.11 ± 24.37**	28.81 ± 24.05
abalone10	0.00 ± 0.00	3.87 ± 11.83	0.00 ± 0.00	**6.18 ± 14.09**	2.75 ± 10.52	*4.82 ± 12.51*

The best performing method is in **bold** face and the second one in *italics*.

nal over-sampling techniques, except MSMOTE, where significant differences are not found. One can observe from this result that the application of MSMOTE is not satisfactory, as this method deteriorates significantly the MAE measure to obtain a non-significant increment in $MMAE$ and GMS. Furthermore, it can be seen that both the results obtained using OGO-ISP and OGO-SP are promising, because similar results (when compared to the SVORIM method) are obtained for MAE while the performance in $MMAE$ is significantly improved. More specifically, the best performing method in $MMAE$ is the OGO-SP procedure, that significantly outperforms

TABLE 4
$MMAE$ mean and standard deviations (Mean $\pm$ SD) obtained over 30 runs by all the methodologies compared.

$MMAE$	SVORIM	MSMOTE	CS-SVORIM	OGO-NI	OGO-ISP	OGO-SP
toy	*0.139 ± 0.059*	0.147 ± 0.073	0.183 ± 0.097	0.156 ± 0.078	**0.137 ± 0.058**	**0.137 ± 0.060**
eucalyptus123vs4vs5	0.524 ± 0.127	0.520 ± 0.119	0.524 ± 0.128	0.519 ± 0.119	*0.517 ± 0.117*	**0.504 ± 0.117**
wisconsin5	2.111 ± 0.434	2.143 ± 0.378	**2.124 ± 0.423**	**2.066 ± 0.370**	2.094 ± 0.345	*2.079 ± 0.385*
eucalyptus1vs2vs345	0.453 ± 0.062	0.440 ± 0.062	**0.395 ± 0.096**	0.432 ± 0.062	0.449 ± 0.066	*0.414 ± 0.070*
wisconsin10	5.117 ± 0.494	5.117 ± 0.634	**4.349 ± 0.821**	5.056 ± 0.430	*4.994 ± 0.496*	5.100 ± 0.448
stock	*0.415 ± 0.109*	0.450 ± 0.122	0.471 ± 0.124	0.421 ± 0.101	0.416 ± 0.111	**0.409 ± 0.102**
newthyroid	0.127 ± 0.094	0.124 ± 0.088	0.120 ± 0.090	0.118 ± 0.099	*0.109 ± 0.096*	**0.099 ± 0.094**
bondrate	*1.783 ± 0.611*	2.021 ± 0.766	**1.633 ± 0.490**	1.817 ± 0.650	*1.817 ± 0.650*	1.883 ± 0.715
housing5	0.450 ± 0.080	0.452 ± 0.084	0.489 ± 0.073	0.457 ± 0.090	**0.431 ± 0.066**	*0.437 ± 0.074*
automobile12vs345vs6	0.350 ± 0.133	0.342 ± 0.112	*0.325 ± 0.130*	**0.314 ± 0.128**	0.348 ± 0.130	*0.325 ± 0.145*
balance-scale	0.109 ± 0.054	*0.107 ± 0.030*	0.123 ± 0.040	0.109 ± 0.050	0.109 ± 0.054	**0.103 ± 0.055**
heating	0.783 ± 0.091	**0.742 ± 0.089**	*0.747 ± 0.099*	0.748 ± 0.096	*0.747 ± 0.093*	**0.742 ± 0.096**
ERA	2.219 ± 0.255	2.188 ± 0.210	2.215 ± 0.258	2.150 ± 0.206	**2.139 ± 0.275**	*2.142 ± 0.289*
auto	0.973 ± 0.212	*0.644 ± 0.243*	0.589 ± 0.206	0.673 ± 0.242	0.713 ± 0.268	**0.615 ± 0.230**
triazines5	*2.400 ± 0.523*	2.593 ± 0.567	**1.980 ± 0.543**	2.575 ± 0.606	2.543 ± 0.579	2.483 ± 0.643
LEV	1.459 ± 0.450	1.352 ± 0.630	**0.810 ± 0.138**	1.333 ± 0.577	*1.247 ± 0.390*	1.271 ± 0.408
housing10	1.247 ± 0.240	1.079 ± 0.324	1.187 ± 0.370	1.067 ± 0.301	*1.049 ± 0.280*	**1.011 ± 0.281**
SWD	1.042 ± 0.095	0.717 ± 0.094	**0.686 ± 0.071**	0.725 ± 0.098	0.721 ± 0.111	*0.716 ± 0.131*
automobile	0.827 ± 0.349	0.834 ± 0.338	**0.795 ± 0.223**	0.804 ± 0.336	0.830 ± 0.338	*0.799 ± 0.328*
ESL12vs3vs456vs7vs89	0.705 ± 0.153	0.591 ± 0.144	**0.548 ± 0.093**	0.583 ± 0.149	*0.577 ± 0.143*	0.591 ± 0.126
machine5	1.088 ± 0.456	1.100 ± 0.463	1.084 ± 0.521	*1.051 ± 0.464*	1.058 ± 0.440	**0.972 ± 0.384**
triazines10	4.967 ± 0.919	4.783 ± 0.784	**4.433 ± 0.817**	*4.550 ± 0.894*	5.200 ± 0.761	4.700 ± 0.651
machine10	2.913 ± 1.264	2.917 ± 1.009	3.020 ± 1.454	2.877 ± 1.242	*2.753 ± 1.213*	**2.570 ± 1.046**
car	0.138 ± 0.062	0.118 ± 0.061	*0.103 ± 0.043*	0.117 ± 0.061	**0.101 ± 0.047**	0.109 ± 0.048
ERA1vs23456vs7vs8vs9	1.660 ± 0.300	1.057 ± 0.161	**1.011 ± 0.072**	1.064 ± 0.133	1.109 ± 0.130	*1.056 ± 0.078*
ESL	1.138 ± 0.418	1.063 ± 0.430	1.148 ± 0.411	**1.026 ± 0.439**	*1.028 ± 0.437*	1.052 ± 0.427
winequality-red	2.103 ± 0.292	1.727 ± 0.441	2.079 ± 0.303	*1.686 ± 0.365*	1.727 ± 0.502	**1.680 ± 0.461**
winequality-white	2.453 ± 0.356	2.476 ± 0.699	2.510 ± 0.858	**2.325 ± 0.551**	2.493 ± 0.775	*2.387 ± 0.617*
abalone5	4.371 ± 1.322	3.668 ± 1.588	**3.221 ± 0.847**	*3.334 ± 1.320*	3.419 ± 1.222	3.446 ± 1.409
abalone10	1.982 ± 0.466	1.313 ± 0.482	**1.038 ± 0.219**	1.199 ± 0.428	1.207 ± 0.391	*1.136 ± 0.327*

The best performing method is in **bold** face and the second one in *italics*.

TABLE 5
Mean ranking results in Acc, MAE and $MMAE$ obtained by all the methods tested and the 30 datasets used.

Ranking	SVORIM	MSMOTE	CS-SVORIM	OGO-NI	OGO-ISP	OGO-SP
Acc	**2.30**	3.48	5.57	3.38	*2.93*	3.33
MAE	**2.35**	3.85	5.47	3.43	*2.65*	3.25
GMS	4.95	3.40	3.73	*3.07*	*3.07*	**2.78**
$MMAE$	4.90	4.37	3.22	*3.15*	3.17	**2.20**
Friedman's test						
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 2.28)$						
F-val.$_{Acc}$: $15.42 \notin C_0$, F-val.$_{MAE}$: $15.53 \notin C_0$						
F-val.$_{GMS}$: $6.14 \notin C_0$, F-val.$_{MMAE}$: $10.70 \notin C_0$						

TABLE 6
Results of the Holm procedure using SVORIM, OGO-ISP and OGO-SP as control methods: corrected α values, compared method and p-values, ordered by the number of comparison (i).

Control alg.: SVORIM		MAE		$MMAE$	
i $\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1 0.0100	0.0200	CS-SVORIM	0.0000_{++}	OGO-SP	0.0000_{--}
2 0.0125	0.0250	MSMOTE	0.0019_{++}	OGO-ISP	0.0002_{--}
3 0.0166	0.0333	OGO-NI	0.0249_{+}	OGO-NI	0.0003_{--}
4 0.0250	0.0500	OGO-SP	0.0624	CS-SVORIM	0.0005_{--}
5 0.0500	0.1000	OGO-ISP	0.5346	MSMOTE	0.2695
Control alg.: CS-SVORIM		MAE		$MMAE$	
i $\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1 0.0100	0.0200	SVORIM	0.0000_{--}	SVORIM	0.0005_{++}
2 0.0125	0.0250	OGO-ISP	0.0000_{--}	MSMOTE	0.0172
3 0.0166	0.0333	OGO-SP	0.0000_{--}	OGO-SP	0.0353
4 0.0250	0.0500	OGO-NI	0.0000_{--}	OGO-NI	0.8902
5 0.0500	0.1000	MSMOTE	0.0008_{--}	OGO-ISP	0.9176
Control alg.: OGO-ISP		MAE		$MMAE$	
i $\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1 0.0100	0.0200	CS-SVORIM	0.0000_{++}	SVORIM	0.0003_{++}
2 0.0125	0.0250	MSMOTE	0.0129_{+}	MSMOTE	0.0129_{+}
3 0.0166	0.0333	OGO-NI	0.1049	OGO-SP	0.0453
4 0.0250	0.0500	OGO-SP	0.2142	CS-SVORIM	0.9176
5 0.0500	0.1000	SVORIM	0.5345	OGO-NI	0.9725
Control alg.: OGO-SP		MAE		$MMAE$	
i $\alpha^*_{0.05}$	$\alpha^*_{0.10}$	Method	p_i	Method	p_i
1 0.0100	0.0200	CS-SVORIM	0.0000_{++}	SVORIM	0.0000_{++}
2 0.0125	0.0250	SVORIM	0.0624_{-}	MSMOTE	0.0000_{++}
3 0.0166	0.0333	MSMOTE	0.2142	CS-SVORIM	0.0353
4 0.0250	0.0500	OGO-ISP	0.2142	OGO-NI	0.0454
5 0.0500	0.1000	OGO-NI	0.3795	OGO-ISP	0.0492

Win ($++$) or lose ($--$) with statistical significant difference for $\alpha = 0.05$
Win ($+$) or lose ($-$) with statistical difference with $\alpha = 0.10$

SVORIM and MSMOTE (without significant differences with respect to the rest of methods). The results obtained from the statistical test can be justified analysing the nature of the developed methods. It is clear that the creation of patterns in the "intra-class" region (as for the method OGO-SP) could expand and make more robust the region assigned to the minority class in the latent target space and therefore this method obtains better $MMAE$ results at a cost of deteriorated MAE values. In contrast, OGO-ISP only creates point in the region within the minority class. This makes this class to receive more attention from the classifier while not damaging the MAE measure. On the other hand, CS-SVORIM is designed to penalise more the errors of patterns belonging to minority classes (note that the ordinal nature of the error is also considered by the formulation of the original SVORIM method) therefore, it helps to improve the $MMAE$ measure with respect to SVORIM, at the cost nevertheless of obtaining worse results for MAE (this suggesting the idea that there two metrics could be con-

flicting objectives when the imbalanced problem is not addressed properly). In this sense, OGO-SP should be preferred in cases when a correct independent ordering of the minority classes in the problem is of vital importance (e.g. in medical applications). On the contrary, if we aim at a balance between a global MAE value and the MAE of the minority classes, then the OGO-ISP should be considered. In either case and as a conclusion, the use of path information in a graph connecting adjacent classes is shown to be very helpful for the over-sampling in imbalanced ordinal datasets and over-sampling can be said to be a more powerful technique for imbalanced cases than a cost-sensitive approach (as suggested in the literature [14]).

5 Conclusions

This paper proposes a first approximation to the problem of over-sampling in imbalanced and ordinal classification problems, which are both common settings for real world datasets. Motivated by the fact that the standard SMOTE algorithm can be seen as an over-sampling performed in the edges of a neighbourhood graph, the proposed approaches are based on extending this graph strategy with the aim of capturing the underlying latent manifold showing the implicit ordering among the classes. This is the first time this graph view of the over-sampling process is given, and it is very convenient for including the necessary ordering constraints in the ordinal regression context. An ordinal cost-sensitive approach is also developed in this paper for comparison purposes. The methods developed show robustness and promising results when compared to the application of the classifier with the original imbalanced distribution and to a standard multiclass over-sampling technique, both for classifying and ordering minority classes. Two main conclusions can be drawn from the study: on the first hand, the fact that the exploitation of the underlying latent manifold via shortest paths is useful to perform the over-sampling process and, on the other hand, the notion that a cost-sensitive approach may in general improve the base performance, but still without reaching the results of over-sampling methods.

As future work, the ensemble proposed in [13], which has been shown to perform well for imbalanced metrics, could be tested in conjunction with the ordinal imbalanced methods developed in this paper. Furthermore, techniques designed to extract the underlying manifold without the construction of a neighbourhood graph could be explored, because a notion of distance is assumed for the neighbourhood graph construction which may not actually hold for the manifold, and the patterns could be over-sampled according to the geodesic distance specified by the constructed manifold. Finally, the idea of over-sampling by convex combination of more than two neighbour patterns (as opposed to only generate new points in the line that passes through two patterns) could be considered in order to enlarge the potential area for synthetic patterns, and non-uniform distributions could be considered for interpolating same-class examples.

Acknowledgments

This work has been subsidized by the TIN2011-22794 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain). Xin Yao's work was supported by an EPSRC grant (EP/J017515/1) and a Royal Society Wolfson Research Merit Award.

References

[1] F. Fernández-Navarro, P. Campoy-Muñoz, M.-D. La Paz-Marín, C. Hervás-Martínez, and X. Yao, "Addressing the EU sovereign ratings using an ordinal regression approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2228–2240, 2013.

[2] M. Pérez-Ortiz, M. Cruz-Ramírez, M. Ayllón-Terán, N. Heaton, R. Ciria, and C. Hervás-Martínez, "An organ allocation system for liver transplantation based on ordinal regression," *Applied Soft Computing*, vol. 14, Part A, no. 0, pp. 88 – 98, 2014.

[3] J. Cardoso, J. F. P. da Costa, and M. Cardoso, "Modelling ordinal relations with SVMs: an application to objective aesthetic evaluation of breast cancer conservative treatment," *Neural Networks*, vol. 18, no. 5-6, pp. 808–817, 2005.

[4] Y. Liu, Y. Liu, S. Zhong, and K. C. Chan, "Semi-supervised manifold ordinal regression for image ranking," in *Proceedings of the 19th ACM international conference on Multimedia (ACM MM2011)*. New York, NY, USA: ACM, 2011, pp. 1393–1396.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[6] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, Oct. 2002.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[8] S. Barua, M. M. Islam, X. Yao, and K. Murase, "Mwmote - majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, no. In press, 2012.

[9] F. Fernández-Navarro, C. Hervás-Martínez, and P. A. Gutiérrez, "A dynamic over-sampling procedure based on sensitivity for multi-class problems," *Pattern Recognition*, vol. 44, no. 8, pp. 1821–1833, Aug. 2011.

[10] S. Wang and X. Yao, "Multiclass Imbalance Problems: Analysis and Potential Solutions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1119–1130, Aug. 2012.

[11] M. Lin, K. Tang, and X. Yao, "Dynamic sampling approach to training neural networks for multiclass imbalance classification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 647–660, 2013.

[12] L. Torgo, R. P. Ribeiro, B. Pfahringer, and P. Branco, "Smote for regression," in *EPIA*, ser. Lecture Notes in Computer Science, L. Correia, L. P. Reis, and J. Cascalho, Eds., vol. 8154. Springer, 2013, pp. 378–389.

[13] M. Pérez-Ortiz, P. A. Gutiérrez, and C. Hervás-Martínez, "Projection based ensemble learning for ordinal regression," *IEEE Transactions on Cybernetics*, vol. Accepted, no. 99, 2013, http://dx.doi.org/10.1109/TCYB.2013.2266336.

[14] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463–484, 2012.

[15] R. Barandela, R. M. Valdovinos, J. S. Snchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Structural, Syntactic, and Statistical Pattern Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3138, pp. 806–814.
[16] W. Chu and S. S. Keerthi, "Support vector ordinal regression," *Neural Computation*, vol. 19, no. 3, pp. 792–815, March 2007.
[17] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21– 27, 1967.
[18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University, 2000.
[19] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," in *International Conference on Artificial Neural Networks*, 1999, pp. 97–102.
[20] A. Shashua and Levin, *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, 2003, vol. 15, ch. Ranking with large margin principle: Two approaches, pp. 937–944.
[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[22] P. Phoungphol, Y. Zhang, Y. Zhao, and B. Srichandan, "Multiclass svm with ramp loss for imbalanced data classification," in *Granular Computing (GrC), 2012 IEEE International Conference on*, Aug 2012, pp. 376–381.
[23] M. Belkin and P. Niyogi, "Towards a theoretical foundation for laplacian-based manifold methods," *J. Comput. Syst. Sci.*, vol. 74, no. 8, pp. 1289–1308, 2008.
[24] J. Sánchez-Monedero, P. A. Gutiérrez, P. Tino, and C. Hervás-Mart ínez, "Exploitation of pairwise class distances for ordinal classification." *Neural Computation*, vol. 25, no. 9, pp. 2450–2485, 2013.
[25] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[26] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of Machine Learning Research*, vol. 6, pp. 1019–1041, 2005.
[27] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li, "Kernel discriminant learning for ordinal regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 906–910, 2010.
[28] A. Fernández, S. García, M. J. del Jesus, and F. Herrera, "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets," *Fuzzy Sets Syst.*, vol. 159, no. 18, pp. 2378–2398, Sep. 2008.
[29] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez, "An Experimental Study of Different Ordinal Regression Methods and Measures," in *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, ser. Lecture Notes in Computer Science, vol. 7209, 2012, pp. 296–307.
[30] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications (ISDA ?09)*, Pisa, Italy, December, 2009 2009, pp. 283–287.
[31] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, no. 0, pp. 21 – 31, 2014.
[32] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

María Pérez-Ortiz was born in Córdoba, Spain. She received her B.S. degree in Computer Science from the University of Córdoba, Spain, in 2011 and her M.Sc. degree in Intelligent Systems from the University of Córdoba, Spain, in 2012. She is currently working towards her Ph.D. degree in the Department of Computer Science and Numerical Analysis (University of Córdoba, Spain), in the area of computer science and artificial intelligence. Her current interests include a wide range of topics concerning machine learning and pattern recognition.

Pedro Antonio Gutiérrez was born in Córdoba, Spain. He received the B.S. degree in Computer Science from the University of Sevilla, Spain, in 2006, and the Ph.D. degree in Computer Science and Artificial Intelligence from the University of Granada, Spain, in 2009. He is currently an Assistant Professor in the Department of Computer Science and Numerical Analysis, University of Córdoba, Spain. His current research interests include patter recognition, evolutionary computation, and their applications.

César Hervás-Martínez was born in Cuenca, Spain. He received the B.S. degree in statistics and operating research from the Universidad Complutense, Madrid, Spain, in 1978, and the Ph.D. degree in mathematics from the University of Seville, Seville, Spain, in 1986. He is currently a Professor with the Department of Computing and Numerical Analysis, University of Córdoba, Córdoba, Spain, in the area of computer science and artificial intelligence and an Associate Professor with the Department of Quantitative Methods, School of Economics. His current research interests include learning algorithms, neural networks, pattern recognition and the modeling of natural systems.

Xin Yao (M'91-SM'96-F'03) is a Chair (Professor) of Computer Science and the Director of the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, Birmingham, U.K. He has authored more than 400 refereed publications in international journals and conferences. His current research interests include evolutionary computation and ensemble learning. Dr. Yao is a Distinguished Lecturer of the IEEE Computational Intelligence Society from 2003 to 2013. He is a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, 2010 BT Gordon Radley Award for the Best Author of Innovation, 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the prestigious Royal Society Wolfson Research Merit Award in 2012, and the 2013 IEEE CIS Evolutionary Computation Pioneer Award. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2003 to 2008. He has been invited to give more than 70 keynote/plenary speeches at international conferences.

We are drowning in information and starving for knowledge.

John Naisbitt

6 Discussion and conclusions

This final part of the thesis includes the main conclusions raised from the previous chapters and outlines some future research lines.

This thesis focus on the resolution of some machine learning challenges associated to the problem of ordinal classification. As stated in the introduction, several goals have been identified: perform a review of the related research, propose new learning methods (specifically a novel but generic decomposition methodology and a kernel learning algorithm), as well as a method able to improve the classification of minority classes in imbalanced environments. This thesis also had as an objective the application of all these new techniques to complex real-world problems. In our humble opinion, these global goals have been achieved. To support this statement, this chapter finalises the thesis with a summary of our contributions, together with some conclusions. We end the chapter with some future research directions. Please note that more details about these conclusions are provided in the corresponding chapters.

6.1. Conclusions

This thesis presents the research performed on the topic of ordinal classification with respect to four main work lines: state-of-the-art analysis, use of decomposition methods for ordinal classification, kernel learning methods and class imbalance. In this section we

summarise the thesis contributions grouped by topics.

6.1.1. State-of-the-art review

The thesis contribution begins with Chapter 2, which performs an exhaustive survey of the ordinal regression methods proposed in the literature. Up to the authors knowledge, there are not similar reviews in this field. This chapter firstly presents the problem setting, clearly differentiating it from other related topics. Then, a taxonomy of ordinal regression methods is proposed, dividing them into four main groups: naïve approaches, binary decompositions, threshold models and augmented binary classification approaches.

We think that the taxonomy presented can assist future researchers or practitioners to choose the best method for a concrete problem, considering also the empirical results provided. It can also help researchers in developing and proposing new techniques, providing a categorisation of current methods. The results presented in all the works associated to this chapter confirm that there is no single method which performs the best in all possible cases and problem requirements. However, several methods could be discarded, especially those presenting the worst performance or a too high computational load. In this sense, naïve approaches (such as the one-vs-one reformulation of the support vector machine [55]) achieve specially good results in terms of accuracy (mainly because of the exhaustive partitioning of all pairs of classes), but not in terms of ordinal measures. Ordinal binary decomposition methods [39, 113] are also a good option in this case, e.g the extreme learning machine version for ordinal classification [33] is an interest option if a low computational cost is the priority. The reformulation of support vector machines to ordinal regression can be considered as the best threshold model, showing competitive results (in terms of ordinal metrics) and computational time values (being these much better than the ones obtained by methods based on artificial neural networks [24, 82]). Moreover, the reduction framework [74] is also an option to consider, given that it obtains a trade-off between accuracy and ordinal metrics. However, the use of the proportional odds model [83] should be restricted, given that it is a linear method and it thus yields worse results in general (recall nonetheless that excluding the machine learning area, the POM and its variants are the most widely used ordinal regression methods [42, 114, 2, 108]).

Concerning the application of ordinal methods to the problem of discriminating the states of development in fish oocytes, similar results are also encountered. For this study, three fish species were considered as well as 5 measures of different nature (two nominal and three ordinal ones). In most cases, ordinal methods exhibit improved robustness and performance compared to nominal techniques (although for some cases nominal methods obtained better results for the nominal metrics). Moreover, the difference between ordinal and nominal techniques has been shown to be higher when the number of states increased,

being this clearly reflected by ordinal quality measures. Finally, the same methods could be highlighted: specially decomposition methods and the ones based on support vector machines.

6.1.2. Decomposition techniques

Chapter 3 of this thesis focus on the use of decomposition methods to tackle the concept of ordinal classification. As noted in the previous subsection, decomposition methods have been shown to perform well for a wide range of datasets [39, 113]. However, how to partition the data and how to fuse the different outputs is still a vividly discussed topic. In this sense, firstly, we have considered and tested several decomposition methods on a battery of datasets and compared them to one of the main proposals of this thesis: the reformulation of the one-versus-all paradigm to ordinal classification. The proposal is based on the computation of several classification models, where each single model was computed to differentiate each class from the remaining ones taking ordinal ranks into account. Probabilistic methods have been selected as base classifiers and the posterior probabilities have been used in conjunction with a ensemble combiner to provide the final output of the classifier. Concerning the results, the advantages of the proposal with respect to the one-versus-all standard paradigm have been confirmed when dealing with ordinal regression. Although multiclass imbalance problems pose important difficulties for machine learning algorithms, this approach also seems to achieve not only good global performance, but also good error rates for all classes independently. Moreover, the proposal has been seen to be scalable (although this is an issue related to the base methodology, it was seen to provide a reasonable time complexity compared to the base method) and interpretable (in the sense that the most determining features for modelling each class can be extracted because it is based on a decomposition strategy).

Decomposition methods have also been used to develop a model to assess the sustainable development (SD) of EU countries. In this case, a trainable combiner [71] (that considers the ordinal nature of the data) is used to fuse the outputs of all models. Firstly, and as a conclusion, the characterisation of the clusters obtained in the previous step of this work reflects a global picture of the SD stage of countries, which could enrich and complement the judgement of stakeholders more than a single indicator score value or trying to find the SD readiness of a country through separate indicators. Secondly, the ordinal regression algorithm proposed is compared to other related classifiers and shows to be competitive yielding better results for this application and supporting the initial assumption of the ordinal nature of clusters defined by the expert and the clustering algorithm. The most determinant variables for the target label have also been studied. These variables are the labour productivity per hour worked, the electricity consumption of households

and the transport greenhouse emissions. In regards to the scenarios, the most important are sustainable consumption, demographic changes, global partnership and climate change and energy, a result that is in line with the three dimensions of SD. On the whole, although it is difficult to assess the direct impact of the indicators on the progress towards sustainability, it can be stated that given the good generalisation performance of the methodology, it may be useful to monitor national strategies for European governments, in a manner similar to that used for rankings (because of the ordinal nature of the clusters obtained), as a managerial tool for supporting decision making and for benchmarking practices to compare results.

In addition, some conclusions are also drawn when considering the use of ordinal classifiers for the design of a donor-recipient matching in organ transplantation. The classification model has been designed in this case to deal with imbalanced and ordinal data to provide a fairer decision maker when allocating an organ to a recipient. The best model obtained from the whole set of methodologies tested, i.e. the proposed decomposition method based on a cascade technique, was used in conjunction with the MELD score, which is the cornerstone of the current allocation policy globally. The experiments show that, although it is a really complex problem which may need more information in order to perform perfectly, the proposal, that is also based on decomposition methods but specifically considering the imbalanced nature of the problem, is able to generalise well on unseen data, helps to avoid draws caused by the MELD score and does seem to work well in more realistic situations. The final rule-based system, which, as said, uses the MELD score and the best performing machine learning model, will consider the allocation of the organ to one of the first recipients in waiting list (these patients being ranked using the MELD score to estimate the patients severity), and the decision is made selecting the patient that presents a higher survival probability. Furthermore, although it has been seen that considering the characteristics of donors and recipients independently can be useful for predicting graft survival (because of the determining factors found in these situations), the use of both sources of information could be even more useful and beneficial for the survival principle.

Finally, a general likelihood-based optimisation framework has been also proposed to better fit the probability distributions obtained for ordered categories when using threshold models such as the ones obtained for the previously discussed ordinal decomposition methods. To do so, a specific probability distribution (log-gamma) is used, which generalises three commonly used link functions (log-log, probit and complementary log-log). The experiments show that the methodology is useful not only to provide a probabilistic output of the classifier but also to improve the performance of threshold models when reformulating the prediction rule to take these probabilities into account. Therefore, its use is advisable in conjunction with the above-mentioned probabilistic decomposition methods,

in order to provide better probability estimations and improve in general the performance of the decomposition ensemble.

6.1.3. Kernel learning

Multi-scale kernels [21] have been barely studied in the literature, although they have been shown to achieve better performance in the presence of heterogeneous attributes [58, 40]. The large number of parameters in multi-scale kernels makes it computationally unaffordable to optimise them by applying traditional cross-validation. Instead, with the aim of better suiting a given dataset, different bounds and strategies have been proposed to optimise this type of kernels [109, 111]. The fist paper in Chapter 4 analyses and compares these alternatives, providing a review of the state-of-the-art in kernel optimisation and some insights into the usefulness of multi-scale kernel optimisation. In this vein, an analytical measure known as centred kernel-target alignment (CKTA) [29, 27] is shown to achieve a good performance and present significant advantages over the rest of methods considered. When applied to the binary support vector machine paradigm, the results show that CKTA with a multi-scale kernel leads to the construction of a well-defined feature space and simpler models, provides an implicit filtering of non-informative features and achieves robust and comparable performance to other state-of-the-art methods even when using random initialisations. Finally, in this paper some considerations about when a multi-scale approach could be, in general, useful are derived and a distance-based initialisation technique for the gradient-ascent method is proposed, which shows promising results.

As said, one of the most widely used ordinal regression algorithms is the proportional odds model (POM) [83], despite the linearity of the resultant decision boundaries. Through different proposals, Chapter 4 also explores the notions of kernel trick and empirical feature space [95, 116] to reformulate the POM method and obtain nonlinear decision boundaries. A new technique is proposed for aligning the kernel matrix taking into account the ordinal problem information (i.e. a reformulation of the above-mentioned CKTA), as well as a regularised gradient ascent methodology which is used to select the optimal dimensionality for the empirical feature space. These proposals can be used to easily kernelise any existing linear ordinal regression method, independently of its formulation. The different experiments show that the proposed kernel techniques are able to increase the performance of linear ordinal regression methods, such as the POM, and reach the results of the state-of-the-art methods, while still being able to derive natural probability estimates.

Finally, a different work on this chapter focus on incorporating privileged information [110] via a kernel function to improve manifold ordinal regression. The paper contri-

butes a new algorithm for combining ordinal regression and manifold learning, based on the idea of constructing a neighbourhood graph and obtaining the shortest path between all pairs of patterns. The paper also proposes to exploit privileged information during graph construction, in order to obtain a better representation of the underlying manifold. The main paper contribution is that this neighbourhood graph can be improved by the use of privileged information, information that is available during training but not in the test phase, and that this information is used for the construction of the kernel matrix, being then a generic method that could be used in conjunction with any kernel technique. When combined with the support vector machine for ordinal classification [26], the results of this paper confirm that privileged information is able to improve generalisation results for almost all the cases considered. As said, the distances used in the kernel matrices are obtained using the privileged features, which (under the assumption that privileged information is really informative) better reflects the data structure.

6.1.4. Imbalanced classification

In Chapter 5 we explore the general idea of synthetic over-sampling in the feature space induced by a kernel function (as opposed to the input space [23, 87, 18, 50]). If the kernel function matches the underlying problem, the classes will be linearly separable and synthetically generated patterns will lie on the minority class region (solving then the main issue of other over-sampling techniques, which make use of the possibly non-linearly separable input space). Since the feature space is not directly accessible, we use the empirical feature space [95] (a Euclidean space isomorphic to the feature space) for over-sampling purposes. The proposed method is framed in the context of support vector machines where imbalanced datasets can pose a serious hindrance for the learning process. The idea of over-sampling in the feature space is investigated in this paper in three scenarios: 1) over- sampling in the full and reduced-rank empirical feature spaces; 2) the use of a flexible kernel learning technique maximising the data class separation to study the influence of the feature space structure (implicitly defined by the kernel function); 3) the definition of a unified framework for preferential over-sampling that spans some of the previous approaches in the literature. From the results of a thorough set of experiments over 50 imbalanced datasets, several conclusions can be drawn from this first study of the chapter: firstly, over-sampling in the empirical feature space is seen to yield better performance than over-sampling in the input space; secondly, the control of the dimensionality of the empirical feature space could lead to better results due to the concentration of spectral properties; thirdly, the kernel used may influence the solution to a great extent, making advisable the optimisation of the feature space structure (although the spherical Gaussian kernel has been shown to perform well for several cases); and finally, that there exist some regions of the dataset which should be preferred for over-sampling and that

multiple kernel learning techniques could be explored in the future with the purpose of over-sampling.

Although standard over-sampling methods can improve the classification of minority classes in ordinal classification, they tend to introduce severe errors in terms of the ordinal label scale, given that they do not take the ordering into account. A specific ordinal over-sampling method is also developed in Chapter 5 for the first time in order to improve the performance of machine learning classifiers. The method proposed includes ordinal information by approaching over-sampling from a graph-based perspective. The results presented in this paper show the good synergy of a popular ordinal regression method (a reformulation of support vector machines [26]) with the graph-based algorithms, and the possibility of improving both classification and ordering of minority classes. The methods developed show robustness and promising results when compared to the application of the classifier with the original imbalanced distribution and to a standard multiclass over-sampling technique, both for classifying and ordering minority classes. Two main conclusions can be drawn from the study: on the first hand, the fact that the exploitation of the underlying latent manifold via shortest paths is useful to perform the over-sampling process and, on the other hand, the notion that a cost-sensitive approach may in general improve the base performance, but still without reaching the results of over-sampling methods.

6.2. Generic discussion and future work

Ordinal classification can be said to be a relatively new area of machine learning with a huge potential for approaching novel and significant applications in medicine, economics, and science in general. It is also interesting for the creation of surrogate models, models intended to substitute the preferences of an user. The main idea behind ordinal classification is that there exist a logical order between the categories, and that this order is of vital importance when handling the data (not only for the classification itself, but also for the generation of new samples or for fitting the parameters of the model). Usually, a latent variable is assumed for ordinal variables (e.g. the age of a person when trying to identify whether that person belongs to one these three categories: infant, teenager or adult). One of the main hypothesis in ordinal regression is that the distance to adjacent classes is lower than the distance to non-adjacent classes. Therefore, it can be said that, ideally, there exists a latent distance-based manifold of the output variable that results in $\mathcal{C}_q$ lying in the space between $\mathcal{C}_{q-1}$ and $\mathcal{C}_{q+1}$. Most ordinal classification methods (and specially threshold models) try to uncover the nature of this assumed underlying outcome. It is clear that the creation of a function that uncovers the path of a general nonlinear manifold (recall that we are referring to a manifold of the output variable) could be difficult.

Because of this, the kernel trick is generally used in conjunction with threshold methods because the kernel function can be able to map the data to a space where the classes are linearly separable. A common approach is to search for $K-1$ parallel hyperplanes [26] (K being the number of classes) for separating the data. However, this could be less flexible than other formulations. A slightly different approach is to decompose the original problem into more simpler classification tasks, relaxing then the order restrictions and the parallelism assumption and obtaining promising results. In this sense, decomposition methods have been seen to be very flexible in this thesis, being their reformulation to other tasks very simple (such as the one considered in this paper, the imbalanced classification problem). A trainable combination rule for the outputs of the different probabilistic models have been seen to be an ideal option, simplifying the task of selecting a previously defined combiner function.

In light of the number of kernel methods specially designed for ordinal classification, it has also been considered as interesting the formulation of a general technique for constructing a kernelised classifier, even when the method can not be cast in terms of dot products. It has been seen that, using such an approach, the pioneer linear methods in ordinal regression are able to obtain similar performance to the most recent advances in this topic. Moreover, as said before, a better way to map the data could be explored to maintain and improve the order information of the classes (i.e. not only use a kernel function but to fit this kernel to the data considering the ordering of the classes). This approach also presents a very good performance in terms of ordinal metrics.

Finally, the imbalance problem has been also seen to be a great handicap for ordinal data. It is clear that there are some classes that are naturally of a lower prior probability (and this problem accentuates when the number of classes grows). It is widely known that the imbalanced problem is not an issue when there is sufficient data. However, this is not the case for most ordinal datasets. Because of this, there is a great need of approaching the imbalanced classification problem, even for binary and multiclass tasks, but specially for this ordinal classification setting. In this vein, the use of kernel functions have been seen to alleviate the problems associated with previous proposals (data inconsistencies that are created due to the way in which new patterns are generated). The use of neighbourhood graphs for over-sampling is also interesting, as it is usually assumed for practical applications that a graph is a good estimation of the real structure of the data and many different graph construction and pruning strategies could be considered. It is also interesting to note that the superiority of over-sampling over cost-sensitive approaches have been demonstrated as well, via different experiments.

From this discussion, there are several main conclusions that can be extracted which mainly sum up to the following statement: there is still a long way to go when referring to the topic of machine learning and specially to ordinal classification. A lot of attention

should be paid to solve the current challenges in order to improve the state-of-the-art and its potential applications. The ordering of the data should be considered for each step of the learning process, not only for the classification method itself but also for generating new patterns, for clustering data, for interpolating missing values, for measuring the performance of different classifiers, and so on, because this has been shown to be very advantageous for a wide range of ordinal datasets.

As future work, several promising lines can be introduced. Firstly, and in line with the previous statement that kernel techniques are common in ordinal classification, a large-scale or under-sampling methodology could be developed, given the high computational load of these methods and their inability to handle large volumes of data. For example, the Nyström approximation [36] could be used to reduce the size of the kernel matrix but taking the ordering of the classes into account (e.g. reducing noisy data).

The decomposition methods developed could also be tested in conjunction with the over-sampling methods proposed in this thesis (as both are designed to handle imbalanced classification). As a more ambitious objective concerning the over-sampling of patterns, some techniques designed to extract the underlying manifold without the construction of a neighbourhood graph could also be explored. This could be interesting because, usually, a notion of distance is assumed for the neighbourhood graph construction which may not actually hold for the manifold. Therefore, the patterns could be over-sampled according to the geodesic distance specified by the constructed manifold, leading then to a much more correct approximation. Moreover, in light of the promising results obtained by over-sampling in the empirical feature space, other methodologies could also be reformulated to work in this ideally linearly separable feature space, such as denoising algorithms or under-sampling ones.

Furthermore, in the context of kernel learning, the over-sampling process could be incorporated in the kernel learning stage or the model training step to search for the most suitable representation of data, not only the better class separation. Finally, other intelligent optimisation techniques could be developed for the generation of synthetic patterns, and pruning strategies could be considered for cleaning the graph representing the patterns (such as the notion of purity of a graph).

Our work on multi-scale kernels also suggest that a distance-based initialisation technique could be used to improve the results of current gradient-descent optimisation strategies, and further research could be done in this line. Nonetheless, our results also encourage the development of a hybrid metaheuristic approach with the gradient ascent method to explore the whole search space and obtain better results. More complex kernel could also be considered, such as the generalised Gaussian kernel with the Mahalanobis distance or other strategies for optimising different kernels in the literature (e.g. opti-

mising a single kernel for the multiple-output classification case or a specific kernel for histograms).

Concerning the application to liver transplantation, a sensitivity analysis could be developed to determine the most important variables for the end-point variable. We also intend to extend the study to other liver transplantation centres in the European Union to unify the procedure and create a more generic supranational organ allocation system. Moreover, the use of privileged information can also be tested in this case, as there are some post-operative factors that could be of vital importance for training and fitting the learning model.

References

[1] A. Agresti. *Analysis of ordinal categorical data*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1984.

[2] A. Agresti. *Categorical Data Analysis*. John Wiley and Sons, 2 edition, 2002.

[3] A. Alshami, A. Lotfi, and S. Coleman. Unified knowledge based economy neural forecasting map. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2012.

[4] K. J. Arrow, P. Dasgupta, L. H. Goulder, K. J. Mumford, and K. Oleson. Sustainability and the measurement of wealth. *Environment and Development Economics*, 17:317–353, 6 2012.

[5] G. Avigad and A. Moshaiov. Interactive evolutionary multiobjective search and optimization of set-based concepts. *Trans. Sys. Man Cyber. Part B*, 39(4):1013–1027, 2009.

[6] M. Babbar-Sebens and B. S. Minsker. Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design. *Applied Soft Computing*, 12(1):182 – 195, 2012.

[7] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri. The imbalanced training sample problem: Under or over sampling? In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 806–814. Springer Berlin Heidelberg, 2004.

[8] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

[9] S. Barua, M. M. Islam, and K. Murase. A novel synthetic minority oversampling technique for imbalanced data set learning. In *International Conference on Neural Information Processing (ICONIP)*, pages 735–744, 2011.

[10] S. Barua, M. M. Islam, X. Yao, and K. Murase. Mwmote - majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, 99(In press), 2012.

[11] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[12] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

[13] P.-M. Boulanger and T. Brechet. Models for policy-making in sustainable development: The state of the art and perspectives for research. *Ecological Economics*, 55(3):337–350, November 2005.

[14] J. Briceño, M. Cruz-Ramírez, M. Prieto, M. Navasa, J. O. de Urbina, R. Orti, M.-Á. Gómez-Bravo, A. Otero, E. Varo, S. Tomé, et al. Use of artificial intelligence as an innovative donor-recipient matching model for liver transplantation: Results from a multicenter spanish study. *Journal of Hepatology*, 61(5):1020–1028, 2014.

[15] J. Briceño, G. Solorzano, and C. Pera. A proposal for scoring marginal liver grafts. *Transplant International*, 13:S249–S252, 2000.

[16] A. M. Brintup, J. Ramsden, and A. Tiwari. An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Computers in Industry*, 58:279–291, 2007.

[17] A. M. Brintup, H. Takagi, A. Tiwari, and J. Ramsden. Evaluation of sequential, multi-objective, and parallel interactive genetic algorithms for multi-objective optimization problems. *Journal of Biological Physics and Chemistry*, 6:137–146, 2006.

[18] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 475–482, Berlin, Heidelberg, 2009. Springer-Verlag.

[19] R. W. Busuttil and K. Tanaka. The utility of marginal donors in liver transplantation. *Liver Transplant*, 9(7):651–663, 2003.

[20] J. S. Cardoso and J. F. P. da Costa. Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research*, 8:1393–1429, 2007.

[21] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[22] S. Chaudhuri and K. Deb. An interactive evolutionary multi-objective optimization and decision making procedure. *Applied Soft Computing*, 10(2):496 – 511, 2010.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[24] J. Cheng, Z. Wang, and G. Pollastri. A neural network approach to ordinal regression. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2008, IEEE World Congress on Computational Intelligence)*, pages 1279–1284. IEEE Press, 2008.

[25] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

[26] W. Chu and S. S. Keerthi. Support vector ordinal regression. *Neural Computation*, 19(3):792–815, March 2007.

[27] C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

[28] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[29] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, 2002.

[30] M. Cruz-Ramírez, C. Hervás-Martínez, J. Fernandez-Caballero, J. Briceño, and M. de la Mata. Multi-Objective Evolutionary Algorithm for Donor-Recipient Decision System in Liver Transplants. *European Journal of Operational Research*, 222(2):317–327, 2012.

[31] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez. Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. *Neurocomputing*, 135:21–31, 2014.

[32] A. L. Dahl. Achievements and gaps in indicators for sustainability. *Ecological Indicators*, 17(C):14 – 19, 2012.

[33] W.-Y. Deng, Q.-H. Zheng, S. Lian, L. Chen, and X. Wang. Ordinal extreme learning machine. *Neurocomputing*, 74(1–3):447–456, 2010.

[34] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[35] L. Diosan, A. Rogozan, and J.-P. Pécuchet. Improving classification performance of support vector machine by genetically optimising kernel shape and hyper-parameters. *Appl. Intell.*, 36(2):280–294, 2012.

[36] P. Drineas and M. W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[37] P. Dutkowski, C. Oberkofler, K. Slankamenac, M. Puhan, E. Schadde, B. Müllhaupt, A. Geier, and P. Clavien. Are there better guidelines for allocation in liver transplantation? A novel score targeting justice and utility in the model for end-stage liver disease era. *Annals of Surgery*, 254(5):745–753, 2011.

[38] S. Feng, N. Goodrich, J. Bragg-Gresham, D. Dykstra, J. Punch, M. DebRoy, S. Greenstein, and R. Merion. Characteristics associated with liver graft failure: The concept of a donor risk index. *American Journal of Transplantation*, 6(4):783–790, 2006.

[39] E. Frank and M. Hall. A simple approach to ordinal classification. In *Proc. of the 12th Eur. Conf. on Machine Learning*, pages 145–156, 2001.

[40] F. Friedrichs and C. Igel. Evolutionary tuning of multiple svm parameters. *Neurocomputing*, 64:107–117, 2004. Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004.

[41] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[42] A. S. Fullerton and J. Xu. The proportional odds with partial proportionality constraints model for ordinal response variables. *Social Science Research*, 41(1):182–198, 2012.

[43] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4):463–484, 2012.

[44] L. García-Hernandez, L. Salas-Morera, and A. Arauzo-Azofra. An interactive genetic algorithm for the unequal area facility layout problem. In *SOCO*, pages 253–262, 2011.

[45] J. Gascón-Moreno, E. G. Ortiz-García, S. Salcedo-Sanz, A. Paniagua-Tineo, B. Saavedra-Moreno, and J. A. Portilla-Figueras. Multi-parametric gaussian kernel function optimization for ε-svmr using a genetic algorithm. In *Proc. of the 11th*

Intern. Conf. on Artificial neural networks, volume 2 of *IWANN'11*, pages 113–120. Springer-Verlag, 2011.

[46] E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, and R. Domínguez-Petit. Exhaustive comparison of colour texture features and classification methods to discriminate cells categories in histological images of fish ovary. *Pattern Recognition*, 46:2391–2407, 2013.

[47] P. A. Gutiérrez, M. Pérez-Ortiz, F. Fernandez-Navarro, J. Sánchez-Monedero, and C. Hervás-Martínez. An Experimental Study of Different Ordinal Regression Methods and Measures. In *7th International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, volume 7209 of *Lecture Notes in Computer Science*, pages 296–307, 2012.

[48] J. Hall, E. Giovannini, A. Morrone, and G. Ranuzzi. A framework to measure the progress of societies. OECD Statistics Working Papers 2010/5, OECD Publishing, July 2010.

[49] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2008.

[50] H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328, 2008.

[51] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[52] G. Hinton and T. Sejnowski. *Unsupervised Learning: Foundations of Neural Computation*. Computational Neuroscience. Mit Press, 1999.

[53] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[54] T. Howley and M. G. Madden. The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review*, 24:379–395, 2005.

[55] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transaction on Neural Networks*, 13(2):415—-425, 2002.

[56] J. C. Hühn and E. Hüllermeier. Is an ordinal class structure useful in classifier learning? *International Journal of Data Mining, Modelling and Management*, 1(1):45–67, 2008.

[57] J. R. Hunter, B. J. Macewicz, N. Lo, and C. A. Kimbrell. Fecundity, spawning and maturity of female Dover Sole, Microstomus pacificus, with an evaluation of assumptions and precision. *Fisheries Bulletin*, 90:101–128, 1992.

[58] C. Igel, T. Glasmachers, B. Mersch, N. Pfeifer, and P. Meinicke. Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 4(2):216–226, Apr. 2007.

[59] J. P. F. J. Stiglitz, A. Sen. Report by the commission on the measurement of economic performance and social progress, 2009.

[60] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, Sept. 1999.

[61] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, Oct. 2002.

[62] I. Jeong and K. Kim. An interactive desirability function method to multiresponse optimization. *European Journal of Operational Research*, 195(2):412–426, 2009.

[63] S. Junquera, E. Román, J. Morgan, M. Sainza, and G. Ramilo. Time scale of ovarian maturation in Greenland halibut (Reinhardtius hippoglossoides, Walbaum). *ICES Journal of Marine Science*, 60:767–773, 2003.

[64] P. Kamath and W. Kim. The Model for End-stage Liver Disease (MELD). *Hepatology*, 45(3):797–805, 2007.

[65] J. Kennedy, A. Gundersen, A. Hoines, and O. Kjesbu. Greenland halibut (Reinhardtius hippoglossoides) spawn annually but successive cohorts of oocytes develop over 2 years, complicating correct assessment of maturity. *Canadian Journal of Fisheries and Aquatic Sciences*, 68:201–209, 2011.

[66] V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

[67] S. B. Kotsiantis and P. E. Pintelas. A cost sensitive technique for ordinal classification problems. In *Proceedings of the Third Hellenic Conference on Artificial Intelligence (SETN2004)*, volume 3025/2004 of *Lecture Notes in Computer Science*, pages 220–229, Samos, Greece, May 5-8 2004.

[68] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, Nov. 2006.

[69] P. Kouvelis, A. A. Kurawarwala, and G. J. Gutierrez. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2):287–303, 1992.

[70] A. Kulig, H. Kolfoort, and R. Hoekstra. The case for the hybrid capital approach for the measurement of the welfare and sustainability. *Ecological Indicators*, 10(2):118 – 128, 2010.

[71] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[72] N. Lee and C. Kirkpatrick. *Sustainable Development and Integrated Appraisal in a Developing World*. Sustainable Development and Integrated Appraisal in a Developing World. Edward Elgar Pub, 2000.

[73] L. Li and H.-T. Lin. Ordinal Regression by Extended Binary Classification. In *Advances in Neural Inform. Processing Syst. 19*, 2007.

[74] H.-T. Lin and L. Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012.

[75] K.-C. Lin. Goodness-of-fit tests for modeling longitudinal ordinal data. *Comput. Stat. Data Anal.*, 54(7):1872–1880, July 2010.

[76] F. Liu, H. Geng, and Y.-Q. Zhang. Interactive fuzzy interval reasoning for smart web shopping. *Applied Soft Computing*, 5(4):433 – 439, 2005.

[77] Y. Liu, Y. Liu, and K. C. C. Chan. Ordinal regression via manifold learning. In W. Burgard and D. Roth, editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11)*, pages 398–403. AAAI Press, 2011.

[78] Y. Liu, Y. Liu, K. C. C. Chan, and J. Zhang. Neighborhood preserving ordinal regression. In *Proceedings of the 4th International Conference on Internet Multimedia Computing and Service (ICIMCS12)*, pages 119–122, New York, NY, USA, 2012. ACM.

[79] Y. Liu, Y. Liu, S. Zhong, and K. C. Chan. Semi-supervised manifold ordinal regression for image ranking. In *Proceedings of the 19th ACM international conference on Multimedia (ACM MM2011)*, pages 1393–1396, New York, NY, USA, 2011. ACM.

[80] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.

[81] S. Marsland. *Machine Learning: An Algorithmic Perspective*. Chapman & Hall/CRC machine learning & pattern recognition series. CRC Press, 2009.

[82] M. J. Mathieson. Ordinal models for neural networks. In J. M. A.-P. N. Refenes, Y. Abu-Mostafa and A. Weigend, editors, *Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, Neural Networks in Financial Engineering, pages 523–536. World Scientific, 1996.

[83] P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society*, 42(2):109–142, 1980.

[84] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monog. on Stat. and Applied Prob. Chapman & Hall/CRC, 2nd edition, 1989.

[85] D. Meadows, D. Meadows, J. Randers, and W. Behrens. *The limits to growth: a report for the Club of Rome's project on the predicament of mankind*. A Potomac Associates Book. Universe Books, 1974.

[86] M. J. Morgan and W. R. Bowering. Temporal and geographic variation in maturity at length and age of Greenland halibut (Reinhardtius hippoglossoides) from the Canadian North-West Atlantic with implications for fisheries management. *ICES Journal of Marine Science*, 54:875–885, 1997.

[87] H. M. Nguyen, E. W. Cooper, and K. Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 3(1):4–21, 2011.

[88] T. Phienthrakul and B. Kijsirikul. Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In *Proceedings of the 2005 conf. on Genetic and evolutionary computation*, GECCO '05, pages 905–911, 2005.

[89] L. Pintér, P. Hardi, P. Bartelmus, I. I. for Sustainable Development, and U. N. D. for Sustainable Development. *Sustainable development indicators: proposals for the way forward prepared for the United Nations Division for Sustainable Development*. International Institute for Sustainable Development, 2005.

[90] E. Rametsteiner, H. Pulzl, J. Alkan-Olsson, and P. Frederiksen. Sustainability indicator development—science or political negotiation. *Ecological Indicators*, 11(1):61 – 70, 2011.

[91] A. Rana, M. A. Hardy, K. J. Halazun, D. C. Woodland, L. E. Ratner, B. Samstein, J. V. Guarrera, R. S. Brown, and J. C. Emond. Survival outcomes following liver transplantation (SOFT) score: a novel method to predict patient survival following liver transplantation. *American Journal of Transplantation*, 8(12):2537–46, 2008.

[92] R. M. Rideout, D. M. Maddock, and M. P. M. Burton. Oogenesis and the spawning pattern in Greenland halibut from the North-west Atlantic. *Journal of Fish Biology*, 54:196–207, 1999.

[93] T. Sato and M. Hagiwara. Idset: Interactive design system using evolutionary techniques. *Computer-Aided Design*, 33(5):367–377, 2001.

[94] D. E. Schaubel, M. K. Guidinger, S. W. Biggins, J. D. Kalbfleisch, E. A. Pomfret, P. Sharma, and R. M. Merion. Survival benefit-based deceased-donor liver allocation. *Am J Transplant*, 9(4 Pt 2):970–81, 2009.

[95] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10:1000–1017, 1999.

[96] A. Shamsheyeva and A. Sowmya. The anisotropic gaussian kernel for svm classification of hrct images of the lung. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004*, pages 439 – 444, 2004.

[97] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *Advances in Neural Information Processing Systems (NIPS)*, pages 937–944. MIT Press, Cambridge, 2003.

[98] C. Soares and P. B. Brazdil. Selecting parameters of svm using meta-learning and kernel matrix-based meta-features. In *Proceedings of the ACM symposium on Applied computing*, pages 564–568, 2006.

[99] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.

[100] C.-J. Su and C.-Y. Wu. Jade implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. *Applied Soft Computing*, 11(1):315 – 325, 2011.

[101] B.-Y. Sun, J. Li, D. D. Wu, X.-M. Zhang, and W.-B. Li. Kernel discriminant learning for ordinal regression. *IEEE Transactions on Knowledge and Data Engineering*, 22:906–910, 2010.

[102] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 39(1):281–288, Feb. 2009.

[103] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme. Cost-sensitive learning methods for imbalanced data. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[104] J. Tompkins, J. White, Y. Bozer, and J. Tanchoco. *Facilities Planning*. Wiley, New York, 4rd ed. edition, 2010.

[105] M.-H. Tseng and H.-C. Liao. The genetic algorithm for breast tumor diagnosis - the case of dna viruses. *Applied Soft Computing*, 9(2):703 – 710, 2009.

[106] UNCED. Agenda 21: The united nations program of action from rio. Technical report, United Nations Conference in Environmental Development, 1992.

[107] UNDP. Sustainability and equity: A better future for all. human development report. Technical report, United Nations Development Program, 2011.

[108] T. Van Gestel, B. Baesens, P. Van Dijcke, J. Garcia, J. Suykens, and J. Vanthienen. A process model to develop an internal rating system: Sovereign credit ratings. *Decision Support Systems*, 42(2):1131–1151, 2006.

[109] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12(9):2013–2036, Sept. 2000.

[110] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5–6):544–557, 2009.

[111] V. N. Vapnik. *Statistical learning theory*. Wiley, 1 edition, Sept. 1998.

[112] J. Verwaeren, W. Waegeman, and B. D. Baets. Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational Statistics & Data Analysis*, 56(4):928–942, 2012.

[113] W. Waegeman and L. Boullart. An ensemble of weighted support vector machines for ordinal regression. *International Journal of Computer Systems Science and Engineering*, 3(1):1–7, 2009.

[114] R. Williams. Generalized ordered logit/partial proportional odds models for ordinal dependent variables. *Stata Journal*, 6(1):58–82, March 2006.

[115] K.-P. Wu and S.-D. Wang. Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. *Pattern Recogn.*, 42(5):710–717, May 2009.

[116] H. Xiong, M. N. S. Swamy, and M. O. Ahmad. Optimizing the kernel in the empirical feature space. *IEEE Transactions on Neural Networks*, 16(2):460–474, 2005.

[117] A. Yardimci. Soft computing in medicine. *Applied Soft Computing*, 9(3):1029 – 1043, 2009.

[118] Z.-Q. Zeng and J. Gao. Improving svm classification with imbalance data set. In *Proc. of the 16th International Conference on Neural Information Processing: Part I*, ICONIP '09, pages 389–398, Berlin, Heidelberg, 2009. Springer-Verlag.