

# A Niching Algorithm to Learn Discriminant Functions with Multi-Label Patterns

J. L. Ávila, E. L. Gibaja, A. Zafra, and S. Ventura

Department of Computer Science and Numerical Analysis. University of Córdoba

**Abstract.** Multi-label patterns are present in many current problems, as protein and gene classification and text categorization, which have become increasing fields of interest. In this paper we present a Gene Expression Programming algorithm for multi-label classification which encodes each individual into a discriminant function that shows whether a pattern belongs to a given class or not. The algorithm also applies a niching technique to guarantee that the population includes functions for each existing class. The algorithm has been compared with some recently published algorithms. The results on several datasets demonstrate the feasibility of this approach to tackle with multi-label problems.

## 1 Introduction

Classification is one of the most studied tasks in the Machine Learning and Data Mining fields. This task basically consists of finding a function which is able to identify the set of an object's attributes (predictive variables) with a label or class identification (categorical variable). This problems are also called single-label problems. Nevertheless, this is not the only possible hypothesis, because numerous problems can be found where a given pattern can be simultaneously mapped to more than one class label. All these problems, which involve assigning all possible proper labels to a given example from a set of prediction variables, are called multi-label classification problems [1].

Some multi-label problems of increasing interest are semantic scene classification [2], text, images and sound categorization [3–6], protein and gene classification [7] and information retrieval [8].

In the literature, many approaches to deal with the problem of multi-label classification can be found. On the one hand, some papers present a problem transformation which pre-process the data set turning a multi-label problem into a single-label one [2, 9]. Another option is to transform the multi-label classification problem into a label ranking task [3]. On the other hand, a specifically designed approach for multi-label data can be carried out [10, 11]. Regarding to the techniques that have been used, it is worth highlighting decision trees [10, 12], bayesian classifiers [13], artificial neural networks [14] and support vector machines [2, 6]. Techniques of lazy learning have also been used, particularly a multi-label version of the well-known K-nearest neighbor algorithm [11], and associative classification methods [15]. Finally it is also worthwhile mentioning

the emerging interest in applying ensemble methods to multi-label classification in order to improve predictions [9, 16] and the development of algorithms for hierarchical multi-label classification [17].

In spite of this great variety of approaches to solve this kind of problems, it seems that multi-label evolutionary approaches have hardly been applied [18], despite that the fact they have solved successfully numerous problems in traditional classification. Therefore, the goal of this paper has been the application and the analysis of this type of algorithms in multi-label problems. We have focused specifically on the Gene Expression Programming [19]. The algorithm developed, called GEP-MLC, encodes a discriminant function in each individual and uses a niching algorithm to guarantee diversity in the solutions.

The paper is organized as follows. The next section introduces the proposed algorithm. Then, the experiments carried out will be described, as will the results of the experiments along a set of conclusions and proposals for future research.

## 2 Algorithm Description

In this section we specify different aspects which have been taken into account in the design of the GEP-MLC algorithm, such as individual representation, genetic operators, fitness function and evolutionary process.

### 2.1 Individual Representation

As mentioned above, the GEP-MLC learns discriminant functions. A discriminant function is a function which is applied to the input features of a pattern (predictive variables) and produces a numerical value associated with the class that the pattern belongs to. To establish this correspondence, a set of thresholds are defined, and intervals of values in the output space are mapped to classification labels. The simplest example is that of the binary classifier, where only one threshold is defined (usually zero). Values to the right of this threshold are associated with patterns belonging to the class, while values to the left will be associated with non-membership in the class.

$$if(f(\mathbf{X}) > 0) \text{ then } \mathbf{X} \in \text{class} \text{ else } \mathbf{X} \notin \text{class} \quad (1)$$

In the case of multi-class problems (the number of classes  $N > 2$ ), there are two approaches to tackle the problem. On the one hand,  $N - 1$  thresholds and  $N$  intervals can be defined. On the other hand,  $N - 1$  functions with only one threshold can be used and deal with the membership of an individual class as a binary classification problem. This last approach is the one that has been used in this study. So, each individual codes in its genotype the mathematical expression corresponding to a discriminant function (binary classifier), and the threshold value of zero has been assigned for all cases. As will be shown later, the class associated to each discriminant function is assigned during the evaluation of the individual.

**Fig. 1.** Conversion from genes to subtrees

Regarding to individual representation, it must be said that in GEP-MLC, as in GEP algorithms, individuals have a dual encoding, that is, they have both genotype and phenotype. Genotype is a lineal string that consists of several genes, whose number and length is fixed and predetermined for each problem. Each gene is divided into two sections, *head* and *tail*. The first one contains terminal and non-terminal elements, whereas the second one can just contain terminal elements<sup>1</sup>. Head length is selected *a priori*, but tail size is calculated as  $t = h(n - 1) + 1$ , where  $t$  is tail size,  $h$  is head length and  $n$  the maximum arity (number of arguments) in the non-terminal set.

Phenotype is an expression tree obtained by a mapping process in which (a) each gene is converted into an expression subtree and (b) subtrees are combined by means of a connection function (in our case, the summation operator - see Figure 1 for an example).

During the evaluation phase, the discriminant function is mapped by a given individual, and its quality as a classifier is calculated for each class defined in the problem. The fitness function used is the F score, that is, the harmonic mean of precision and recall [20]:

$$fitness = \frac{2 \times precision \times recall}{precision + recall} \quad (2)$$

In contrast with single-label algorithms, that assign the label that produces the highest fitness value, the GEP-MLC algorithm calculates a fitness value for each label, storing  $N$  raw fitness values (one per label). As will be shown, the fitness value used in the selection phase is obtained by transforming these values during the token competition phase and taking the highest one.

## 2.2 Evolutionary Algorithm

The structure of the algorithm is similar to that of the standard GEP algorithm. Thus, GEP-MLC uses all the genetic operators defined in the standard GEP algorithm, that is, crossover, transposition and mutation. The detailed description of such operators is described in the original paper of Ferreira [19].

Algorithm 1 shows the pseudo-code of the GEP-MLC algorithm. As can be seen, it computes the  $N$  fitness values, one for each label. In addition, the algorithm implements the *Token Competition* technique to correct individual fitness after its evaluation.

Token competition [21] is used in classification algorithms to emulate the niching effect, as shown in natural ecosystems. So, for each positive pattern and label, a *token* is a stake which is won by the individual with the highest

---

<sup>1</sup> In this context, terminal elements are functions without arguments, and non-terminal ones are functions with one or more arguments.

---

**Algorithm 1** GEP-MLC pseudo-code

---

```
Generate initial population  $P(0)$ 
 $g_{count} \leftarrow 0$ 
while  $g_{count} < g_{max}$  do
  for all  $l_i \in L$  (labels set) do
    Evaluate all individuals in  $P(g_{count})$ 
  end for
  for all  $l_i \in L$  do
     $total\_tokens =$  number of patterns belonging to  $l_i$ 
     $tokens\_won_{Ind_j} = 0 \forall Ind_j \in P(g_{count})$ 
    for all positive pattern (token) do
       $tokens\_won_{Ind_j} ++$  where  $Ind_j$  has the highest  $fitness_{l_i}$  and correctly classify the pattern
    end for
    Update  $fitness_{l_i}(\forall Ind_j \in P(g_{count}))$  by using formula 3
  end for
  Do parents selection
  Apply genetic operators
  Update population
   $g_{count} ++$ 
end while
Generate classifier
```

---

fitness correctly classifying the pattern. When all the tokens are distributed, the algorithm corrects the fitness of each individual using the expression

$$new\_fitness = \frac{original\_fitness \times tokens\_won}{total\_tokens} \quad (3)$$

Token Competition penalizes individuals that, despite their average fitness, do not contribute to the classifier. On the other hand, it helps both the individuals with good fitness that correctly classify many patterns, and individuals specialized in classifying strange patterns, which are not usually correctly classified as the best individuals. In the proposed algorithm, there will be as many token competitions as labels in the training set. Thus, each token will be associated with a certain label and will be played by an individual with the fitness associated to such label. When the algorithm finishes, only individuals that have won any token will be in the learned classifier.

### 3 Experimental Section

Experiments carried out have compared the performance of the proposed GEP-MLC algorithm to other multi-label classification ones. This section explains several details related with these experiments such as data sets and algorithmic details.

For the experimentation, the algorithm proposed has been tested with six multi-label data sets, *scene*, *emotions*, *yeast*, *genbase*, *mediamill* and *TMC2007.Scene*

data set contains a series of patterns about kinds of landscapes while Emotions data set is concerned with the classification of songs according to the emotions they evoke. Yeast and genbase include information about protein function. Mediamill data set consists of patterns about multimedia files. Finally, TCM2007 data set contain information about text classification. Nominal attributes of datasets has been binarized as in [22].

Table 1 resumes the main features (number of classes and patterns) of each data set. It includes also to measures about how much multi-label is a data set: Label cardinality (the average number of labels per example) and label density (the same number divided by the total number of labels) [1, 9].

All data sets have been randomly split into 10 partitions in order to carry out a 10-fold cross validation. For each test, 3 different runs have been executed and an average value has been calculated in order to measure the performance of the evolutionary algorithm as independently as possible from its randomness.

<b>Dataset</b>	<b>#patterns</b>	<b>#labels</b>	<b>Cardinality</b>	<b>Density</b>
Scene	2407	6	1.061	0.176
Genbase	662	27	1.252	0.046
Emotions	593	6	1.868	0.311
TMC2007	28596	22	2.158	0.098
Yeast	2417	14	4.228	0.302
Mediamill	43907	101	4.376	0.043

**Table 1.** Features of the data sets

A set of tests was made to find the optimal parameters of the algorithm. And so, we have used 6 genes per individual with a head size of 35. The population size has been 1000 individuals with 60 generations. The tournament size is 3 and the probabilities of mutation, crossover and transposition have a value of 0.2, 0.7 and 0.4 respectively.

The classical measures of accuracy (acc), precision (prec) and recall (rec) have been extended from single-label to multi-label to compare these methods. Thus, we have used the macro-averaged approach proposed in [23], where precision and recall are first evaluated locally for each category, and then globally by averaging over the results of the different categories.

GEP-MLC implementation was made using the JCLEC library [24]. The rest of the algorithms used in the tests were available in the MULAN library<sup>2</sup>. This is a Java package built on top of the WEKA data mining tool [25] which contains several problem transformation and algorithm adaptation methods for multi-label classification, an evaluation framework that computes several evaluation measures and a class providing data set statistics.

<sup>2</sup> MULAN is freely available at <http://mlkd.csd.auth.gr/multilabel.html>

<b>Algorithm</b>	<b>Binary Relevance</b>			<b>Label Powerset</b>			<b>ML-Knn</b>			<b>GEP-MLC</b>		
<b>Dataset</b>	acc	prec	rec	acc	prec	rec	acc	prec	rec	acc	prec	rec
Scene	0.538	0.630	0.623	0.587	0.594	0.597	0.647	0.799	0.675	0.709	0.746	0.744
Genbase	0.634	0.550	0.596	0.621	0.535	0.533	0.585	0.428	0.383	0.755	0.650	0.582
Emotions	0.203	0.280	0.597	0.290	0.276	0.285	0.126	0.321	0.029	0.903	0.724	0.695
TMC2007	0.443	0.582	0.503	0.613	0.603	0.573	0.402	0.437	0.483	0.543	0.618	0.540
Yeast	0.141	0.192	0.129	0.131	0.193	0.192	0.113	0.114	0.113	0.738	0.715	0.649
Mediamill	0.582	0.609	0.499	0.594	0.556	0.521	0.470	0.419	0.127	0.703	0.669	0.581

**Table 2.** Experimental results

## 4 Results and Discussion

The performance of the proposed algorithm has been compared to other methods for multi-label classification, namely, Binary Relevance, Label Powerset and the ML-KNN method [11]. Binary Relevance and Label Powerset uses C4.5 as a base classifier. Table 2 shows the experimental results.

As can be observed, the proposed algorithm obtains, in general, good results which are comparable, even better in some cases, to the results of the other studied algorithms.

It can also be observed that the differences between scores of GEP-MLC and these algorithms are increased when the data set has more cardinality, in other words, more multi-label features. Thus, with data sets as scene, TMC, genbase and mediamill, whose cardinality is close to one (nearly a single-label problem), GEP-MLC algorithm obtains better recall results, but they are comparable to those obtained by ML-KNN and LP. With respect to the accuracy, scores of GEP-MLC are also better except for the TMC data set. Such result would be due to the fact that TMC is a data set with boolean features which have been converted into numbers in order to deal with discriminant functions.

In contrast, when the results for emotions are analyzed, GEP-MLC is found to obtain better scores for accuracy, precision and recall measures, finding they are quite better than the scores of the studied algorithms. Furthermore, the same result can be observed with the yeast data set, the one with the highest number of labels and values of density and cardinality (the most multi-label data set).

The noticeable differences between GEP-MLC and the rest of algorithms in these data sets can be understood because Binary Relevance and Label Powerset preprocess the training set to turn it into a single-label one. This preprocessing could diminish the algorithm performance if the dataset is highly multi-label.

## 5 Conclusions and Future Work

This study presents the GEP-MLC algorithm, an evolutionary algorithm for multi-label classification. This algorithm, based on GEP, codifies discriminant functions that indicate that a pattern belongs to a certain class in such a way

that the final classifier is obtained by combining several individuals from the population. It uses a niching technique (token competition) to ensure that the population will present functions representing all the classes present in a given problem. Studies have been carried out to check the performance of our algorithm and compare it with those of other available algorithms, to verify that GEP-MLC renders the best performance in terms of accuracy, precision and recall, and that it is, at the same time, much less insensitive to the degree of overlapping in its classes, which is a very positive characteristic. Regarding to future research, the algorithm is being tested in other domains and, besides, compared with other approaches for multi-label classification such as SVM or decision trees. In addition the proposed model will be extended to manage with hierarchical classification or ranking problems which are closely related to multilabel problems.

**Acknowledgment** This work has been financed in part by the TIN2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (CICYT), the P08-TIC-3720 project of the Andalusian Science and Technology Department and FEDER funds.

## References

1. Tsoumakas, G., Katakis, I.: Multi label classification: An overview. *International Journal of Data Warehousing and Mining* **3**(3) (2007) 1–13
2. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. *Pattern Recognition* **37**(9) (September 2004) 1757–1771
3. Loza, Fürnkranz, J.: Efficient pairwise multilabel classification for large-scale problems in the legal domain. (2008) 50–65
4. Chang, Y.C., Chen, S.M., Liao, C.J.: Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. *Expert Systems with Applications* **34**(3) (2008) 1948–1953
5. Jiang, A., Wang, C., Zhu, Y.: Calibrated rank-svm for multi-label image categorization. In: *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on.* (2008) 1450–1455
6. Li, T., Ogihara, M.: Detecting emotion in music. In: *Proceedings of the 14th international conference on music information retrieval (ISMIR03), Baltimore, USA* (2003)
7. Jung, J., Thon, M.R.: Gene function prediction using protein domain probability and hierarchical gene ontology information. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on.* (2008) 1–4
8. Sarinapakorn, K., Kubat, M.: Induction from multi-label examples in information retrieval systems: A case study. *Applied Artificial Intelligence* **22**(5) (2008) 407–432
9. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multi-label classification. *LNCS* **4701** (2007) 406–417
10. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. *LNCS* **2168** (2001)
11. Zhang, M.L., Zhou, Z.H.: A k-nearest neighbor based algorithm for multi-label classification. Volume 2., *The IEEE Computational Intelligence Society* (2005) 718–721

12. Noh, H.G., Song, M.S., Park, S.H.: An unbiased method for constructing multilabel classification trees. *Computational Statistics & Data Analysis* **47**(1) (2004) 149–164
13. Ghamrawi, N., Mccallum, A.: Collective multi-label classification. In: *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, New York, USA, ACM Press (2005) 195–200
14. Zhang, M.L., Zhou, X.H.: Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering* **18**(10) (October 2006) 1338–1351
15. Rak, R., Kurgan, L., Reformat, M.: A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation. *Data & Knowledge Engineering* **64**(1) (January 2008) 171–197
16. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. *Machine Learning* **39**(2/3) (2000) 135–168
17. Vens, C., Struyf, J., Schietgat, L., Džeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. *Machine Learning*
18. Vallim, R.M.M., Goldberg, D.E., Llorà, X., Duque, T.S.P.C., Carvalho, A.C.P.L.F.: A new approach for multi-label classification based on default hierarchies and organizational learning. In: *GECCO '08: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*. (2008) 2017–2022
19. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. *Complex Systems* **13**(2) (2001) 87–129
20. Han, J., Kamber, M.: *Data Mining: Methods and Techniques*. Second edn. Morgan Kaufmann (2006)
21. Wong, M.L., Leung, K.S.: *Data Mining Using Grammar-Based Genetic Programming and Applications*. Genetic Programming Series. Kluwer Academic Publishers (2002)
22. Zhou, C., Xiao, W., Tirpak, T.M., Nelson, P.C.: Evolving accurate and compact classification rules with gene expression programming. *IEEE Trans. Evolutionary Computation* **7**(6) (2003) 519–531
23. Sebastiani, F.: Machine learning in automated text categorization. *ACM Comput. Surv.* **34**(1) (March 2002) 1–47
24. Ventura, S., Romero, C., Zafra, A., Delgado, J.A., Hervás, C.: JCLEC: A Java framework for evolutionary computation. *Soft Computing* **12**(4) (2008) 381–392
25. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*. Second edn. Morgan Kaufmann (2005)