Algoritmos Evolutivos para Descubrimiento de Reglas de Predicción en la Mejora de Sistemas Educativos Adaptativos basados en Web

Cristóbal Romero, Sebastián Ventura, Carlos de Castro, Enrique García

Departamento de Informática y Análisis Numérico. Universidad de Córdoba
14071, Campus Rabanales
{cromero,sventura,cdcastro,egarcia}@uco.es

Resumen: Este artículo muestra la utilización de los algoritmos evolutivos para el descubrimiento de reglas de predicción que se utilizarán en la mejora de Cursos Hipermedia Adaptativos basados en Web. Se ha desarrollado una herramienta de minería de datos específica para descubrir relaciones entre los datos de utilización recogidos durante las ejecuciones de los distintos alumnos. Esta información puede ser de gran utilidad para el profesor o autor del curso, para la toma de decisiones sobre qué modificaciones son las más adecuadas para mejorar el aprendizaje de los alumnos. Para la realización de la búsqueda de reglas de predicción se ha utilizado programación genética basada en gramáticas multi-objetivo y se han comparado con algoritmos clásicos de descubrimiento de reglas.
Palabras clave: Algoritmos evolutivos, minería de datos, reglas de predicción, descubrimiento de conocimiento, cursos hipermedia adaptativos, enseñanza basada en web.

Abstract: In this paper we show the use of evolutionary algorithms for discovering prediction rules to improve web-based adaptive hypermedia courses. We have developed a specific data mining tool to discover relationship between the usage data pickup during the execution of different students. This information can be very useful to the courseware author in order to make decisions about what are the most appropriated modifications to improve the learning of the students. In order to do prediction rule discovering we have used multi-objective grammar-based genetic programming and we have compared it with other classic algorithm for rule discovering.
Key words: Evolutionary algorithms, data mining, prediction rules, knowledge discovery, adaptive hypermedia courses, web-based education.

1. Introducción

En los últimos años hemos asistido a un crecimiento exponencial de la aplicación de las tecnologías basadas en web e inteligencia artificial a los sistemas de educación a distancia, lo cual ha dado lugar a la aparición de los Sistemas Educativos Adaptativos basados en Web [Brusilovsky 98] que permiten adaptar la enseñanza en Internet a cada alumno en particular dependiendo de su nivel de conocimiento. Estos sistemas almacenan una enorme cantidad de información, que podría utilizarse en la mejora del propio sistema. Sin embargo, hasta la fecha existen muy pocas aplicaciones en las que se utilicen técnicas para el descubrimiento de información que permita su propia mejora [Zaïene 01]. Este tipo de técnicas ayudarían al profesor a evaluar el aprovechamiento del sistema por parte de los estudiantes: secuenciación de contenidos, validación de actividades, sistemas de calificación automática, etc. Estas técnicas de minería de datos o data mining (DM) ya se han aplicado con éxito en sistemas de comercio electrónico o e-commerce, para comprender el comportamiento de clientes en línea de sistemas de

comercio electrónico y poder incrementar las ventas [Srivastava et al. 00]. Para conseguir su objetivo, las herramientas de DM utilizan técnicas de extracción de conocimiento para descubrir información útil en la mejora del sistema. Aunque los métodos de descubrimiento de información utilizados en ambas áreas (e-commerce y e-learning) son similares, los objetivos finales tienen matices totalmente diferentes debido a que en e-commerce el objetivo es guiar a los clientes durante la compra para maximizarla, mientras que en e-learning el objetivo es guiar a los estudiantes durante su aprendizaje para maximizarlo. Por lo tanto, cada uno tiene unas características específicas que requieren de un tratamiento diferente dentro del problema de minería de Web.

La aplicación de técnicas de minería de datos en educación se puede ver desde dos puntos de vista u orientaciones distintas:

- Orientado hacia los autores. Con el objetivo de ayudar a los profesores y/o autores de los sistemas de e-learning para que puedan mejorar el funcionamiento o rendimiento de estos sistemas a partir de la información de utilización de los alumnos. Sus principales aplicaciones son: obtener una mayor realimentación de la enseñanza, conocer más sobre como los estudiantes aprenden en el web, evaluar a los estudiantes por sus patrones de navegación, reestructurar los contenidos del sitio web para personalizar los cursos, clasificar a los estudiantes en grupos, etc.
- Orientado hacia los estudiantes. Con el objetivo de ayudar o realizar recomendaciones a los alumnos durante su interacción con el sistema de e-learning para poder mejorar su aprendizaje. Sus principales aplicaciones son: sugerir buenas experiencias de aprendizaje a los estudiantes, adaptación del curso según el progreso del aprendiz, ayudar a los estudiantes dando sugerencias y atajos, recomendar caminos más cortos y personalizados, etc.

En el presente artículo vamos a proponer una metodología que permite que los profesores o autores puedan mejorar los sistemas educacionales adaptativos basados en web utilizando la información descubierta a partir de los datos de utilización de los alumnos. La minería de datos o data mining [Zytkow et al. 01] es una de las áreas de investigación que ha experimentado un crecimiento más espectacular en los últimos años, ofreciendo herramientas potentes para el análisis de grandes bases de datos utilizadas en la empresas, industrias y ciencias. Aunque el término minería de datos se suele identificar con el concepto completo de descubrimiento de conocimiento, en realidad es sólo una parte dentro del proceso de descubrimiento de conocimiento que, según Agrawal, [Agrawal et al. 94] se define como "el proceso no trivial para identificar conocimiento válido, novedoso y potencialmente útil" y es un proceso iterativo que consta de las fases de preprocesado, minería de datos y post-procesado. Una de las aplicaciones más importantes de la minería de datos es el descubrimiento de reglas de asociación [Hipp et al. 00] en el que se identifican relaciones entre elementos que suelen aparecer conjuntamente en un dominio determinado. Existen diferentes aproximaciones a este problema, dando lugar a distintos tipos de reglas de asociación: reglas de asociación cuantitativas, reglas de asociación negativas, reglas causales, reglas generalizadas o de predicción, etc. El descubrimiento de reglas de asociación se ha planteado frecuentemente en sistemas de comercio electrónico, en los que se han buscado reglas del tipo: "Si un cliente compra el producto x1 y x2, entonces comprará el producto x3 con una determinada probabilidad". Debido a esta aplicabilidad y a su comprensión inherente, el establecimiento de reglas de asociación se ha convertido en un método muy popular de minería de datos. Sin embargo, los métodos clásicos empleados en esta tarea [Zytkow et al. 01] presentan algunos problemas, como:

- La complejidad del algoritmo y el número de reglas crece exponencialmente con el número de elementos.
- Las reglas interesantes se deben escoger de las reglas generadas. Esto puede ser costoso, ya que las reglas generadas suelen ser muchas y las útiles son un porcentaje pequeño.

Una forma muy prometedora de solucionar estos problemas es utilizando algoritmos evolutivos para el descubrimiento de reglas [Freitas 02] dado que éstos realizan una búsqueda global y tratan mucho mejor la interacción entre los datos que los algoritmos clásicos de inducción de reglas. Este trabajo propone el descubrimiento de reglas para la obtención de información útil para el profesor en forma de

relaciones existentes entre datos de utilización. Hemos utilizando un algoritmo evolutivo multi-objetivo en el que además se pueden indicar restricciones que deben cumplir las reglas en base a distintas consideraciones particulares de cada usuario. El trabajo se organiza de la siguiente forma: Primero vamos a describir algunos antecedentes al problema que hemos abordado. Seguidamente, introduciremos el problema del descubrimiento de reglas de predicción, y describiremos nuestra metodología de mejora de cursos. Posteriormente, detallaremos el algoritmo evolutivo específico utilizado para la búsqueda de reglas de predicción. En la sección experimental, describiremos las pruebas y resultados obtenidos. Por último, se enumerarán las principales conclusiones y futuro trabajo.

2. Antecedentes

La minería de datos es el proceso de descubrimiento de conocimiento para encontrar información no trivial, previamente desconocida y potencialmente útil de grades repositorios de datos [Zytkow et al. 01]. Un caso particular de la minería de datos es la minería de web o web mining [Scime 04], que como el propio nombre indica consiste en la aplicación de técnicas de minería de datos para extraer conocimiento a partir de datos de la Web. Se pueden distinguir tres tipos de minería de Web:

- Minería de contenidos web. Es el proceso de extraer información a partir de los contenidos de los documentos Web.
- Minería de estructura web. Es el proceso de descubrir información a partir de la estructura de la Web.
- Minería de utilización web. Es el proceso de descubrir información a partir de los datos de utilización de la Web.

De los estos tres tipos de minería de web, el que más se ha utilizado para el descubrimiento de información en los sistemas de enseñanza basada en web es la minería de utilización web o web usage mining. A continuación se describen algunos trabajos específicos de aplicación de minería de datos en e-learning, en concreto utilizando técnicas de descubrimiento de reglas.

Uno de los pioneros de la utiliza técnicas de minería web en sistemas de e-learning es Osmar Zaïne [Zaïene 01] y en trabajos más recientes propone utilizar agentes recomendadotes [Zaïene 02] para recomendar actividades de aprendizaje en línea o atajos en un curso web basándose en los historiales de acceso y mejorar el proceso de aprendizaje en línea. Concretamente utiliza minería de reglas de asociación para entrenara al agente recomendador y construir un modelo que representa el comportamiento de acceso o asociaciones entre actividades de aprendizaje en línea. Otro trabajo que analiza los ficheros log de entornos de aprendizaje web utilizando técnicas de minería de reglas de asociación y filtrado colaborativo, es el realizado por Feng-Hsu Wang [Wang 02] para descubrir patrones de navegación útiles y proponer un modelo de navegación. El modelo de navegación consiste en dos tipos de relaciones: relaciones de asociación y relaciones de secuencia entre documentos. También se están utilizando técnicas de softcomputing, por ejemplo Pao-Ta Yu y otros [Yu et al. 01] proponen la utilización de reglas de asociación difusas para descubrir relaciones entre patrones de comportamiento de los estudiantes, incluyendo el tiempo de acceso, números de páginas leídas, preguntas contestadas, mensajes leídos y enviados, etc. Finalmente un trabajo que emplea algoritmos evolutivos es el realizado por Behrouz Minaei-Bidgoli y William F. Punch [Minaei-Bidgoli et al. 03] para realizan un análisis de asociación para predecir el rendimiento de los estudiantes. Utilizan clustering de recursos web valorados y descubrimiento de reglas de asociación interesantes mediante algoritmos genéticos para optimización de minería de datos con el objetivo es clasificar a los estudiantes para predecir su clasificación final basándose en las características extraídas de los ficheros logs.

3. Reglas de Predicción

El modelado de dependencias [Zytkow et al. 01], también denominado por algunos autores como inducción de reglas de predicción o reglas generalizadas tiene como cuyo objetivo el descubrimiento de reglas interesantes para mostrárselas al usuario. Estas reglas, que representan relaciones de dependencia importantes entre los datos

y que se pueden utilizar para la posterior toma de decisiones, presentan el siguiente formato:

SI $Cond_1$ **Y** … $Cond_i$ … **Y** $Cond_m$ **ENTONCES** Pred

donde cada condición Condi y la predicción de la regla Pred están formados por una tripleta: (Atributo, Operador, Valor). Este tipo de reglas muestra la relación existente entre el antecedente, que contiene las condiciones sobre los valores de los atributos predictores, y el consecuente, que contiene la predicción sobre el valor del atributo objetivo.

Llegado este punto, habría que puntualizar la diferencia que existe entre el descubrimiento de reglas de predicción y el descubrimiento de reglas de asociación [Freitas 00], una tarea similar pero algo más general, en la que el objetivo es la búsqueda de todas las posibles relaciones entre atributos y donde puede haber incluso varios atributos en el consecuente de la regla. La tarea de descubrimiento de reglas de asociación, introducida por Agrawal [Agrawal et al. 94] que la define como el problema de encontrar todos los elementos que son frecuentes con respecto a un umbral mínimo de soporte y confianza. El soporte indica el porcentaje de instancias que contienen tanto consecuente como el antecedente y la confianza indica el porcentaje de instancias que contienen el consecuente también contienen al antecedente. De forma, que el usuario debe especificar un valor mínimo de la medida de soporte y confianza, siendo el objetivo de la tarea encontrar todas las reglas que superen esos valores. Una regla de predicción puede verse como una regla de asociación con un solo elemento en su consecuente, por lo que cualquier algoritmo de descubrimiento de reglas de asociación se puede modificar fácilmente para añadir esta restricción y descubrir sólo reglas de predicción. Hemos utilizado reglas de predicción en lugar de reglas de asociación debido a que se pueden utilizar más fácilmente para la toma de decisiones, son intuitivamente más comprensibles y muestran relaciones directas entre los elementos en lugar de todas las posibles relaciones.

4. Metodología Propuesta

La metodología de mejora de Cursos Hipermedia Adaptativos basados en Web que proponemos [Romero et al. 04] es una metodología dinámica y cíclica (ver Figura 1) de forma que mientras más se ejecute el curso por los alumnos, más información se dispondrá para poder mejorar el curso. En esta metodología podemos distinguir cuatro etapas o fases principales:

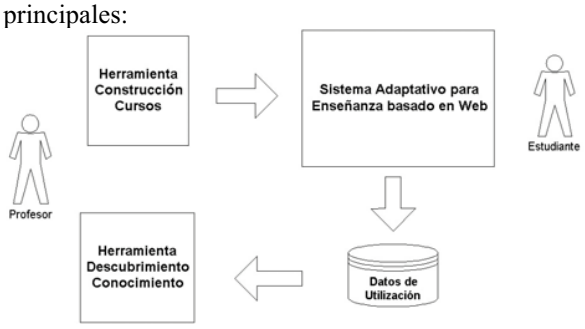


Figura 1. Metodología propuesta para la mejora de cursos utilizando minería de datos.

- **Construcción del curso.** Es la primera etapa y es donde se construye el curso. El profesor suele ser el encargado de construir el curso adaptativo proporcionando toda la información tanto de contenido como de estructura necesaria para el curso. Normalmente se suele utilizar una herramienta autor genérica o específica para facilitar esta tarea [Brusilovsky 98]. Al finalizar esta etapa el curso debe de ser publicado en un servidor web para que los alumnos puedan utilizarlo de forma remota.
- **Ejecución del curso.** Los estudiantes utilizando un navegador web se deben de conectar al servidor web donde se encuentra localizado el curso para poder realizarlo. Mientras los alumnos ejecutan el curso de forma transparente se va recogiendo información de utilización y esta se va almacenando en el servidor en los distintos ficheros logs.
- **Descubrimiento Reglas de Predicción.** Tras pasar los ficheros logs a una base de datos el profesor aplica el algoritmo evolutivo para el descubrimiento de reglas de predicción y obtener relaciones importantes entre todos los datos de utilización recogidos. Para facilitar esta tarea se ha desarrollado una herramienta específica (que hemos denominado EPRules) orientada al profesor.
- **Mejora del curso.** El profesor utilizando la información que le proporciona las relaciones descubiertas realiza las modificaciones que crea

más adecuadas para mejorar el rendimiento del curso. Para ello analiza las reglas y utiliza de nuevo la herramienta autor para realizar los cambios oportunos en el curso en el modelo del dominio (cambiando o añadiendo contenidos), el modelo pedagógico (cambiando, añadiendo o eliminando reglas) y el módulo interfaz (cambiando partes del interfaz).

Nuestro objetivo es, por tanto, descubrir información relevante desde el punto de vista didáctico y de la efectividad de la enseñanza en forma de reglas a partir de estos datos de seguimiento almacenados para todos los alumnos que ejecutan el curso. En la siguiente sección se mostrará el algoritmo evolutivo empleado para el descubrimiento de esta información.

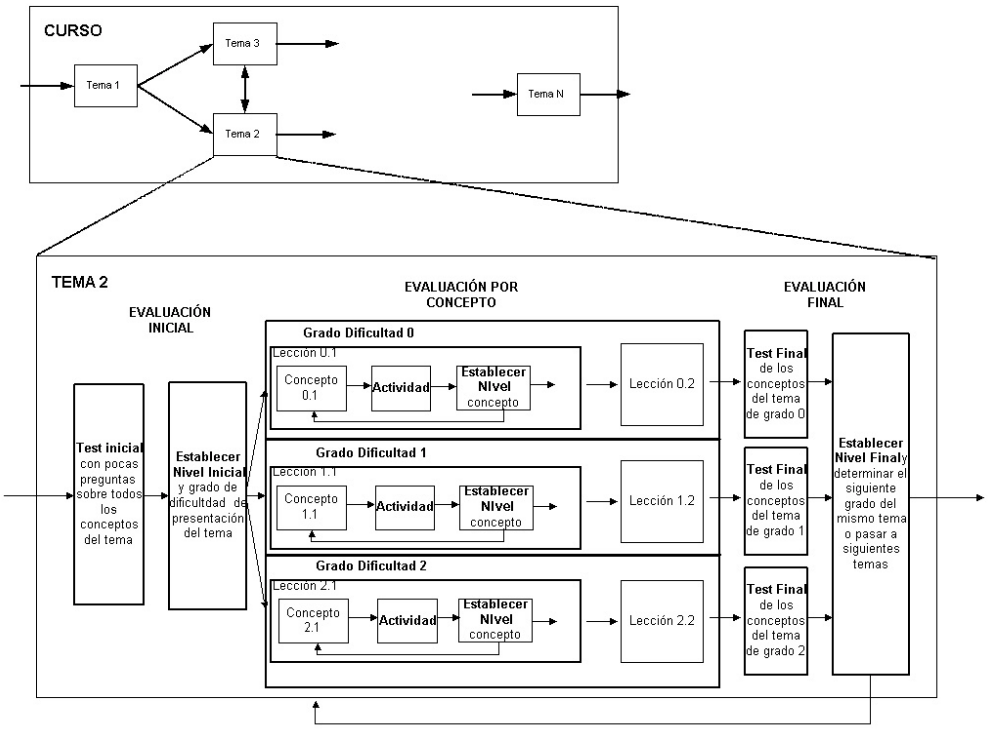


Figura 2. Funcionamiento del sistema AHA! modificado

5. Datos de Utilización

Para poder capturar la información de utilización necesaria para realizar el descubrimiento de reglas de predicción hemos desarrollado un curso hipermedia adaptativos basados en web sobre el sistema operativo Linux realizado con la arquitectura genérica AHA [De Bra et al. 00]. Hemos escogido el sistema AHA debido a que además de ser un modelo genérico de sistema hipermedia adaptativo, captura toda la información de utilización de los usuarios y se dispone del código fuente para poder realizar modificaciones. Hemos modificado el sistema AHA para potenciarlo en educación [Romero et al. 02] permitiendo adaptar el contenido del curso en tres niveles de dificultad dependiendo de los aciertos obtenido por el alumno en los distintos test iniciales, finales y actividades del curso (Ver figura 2).

La Figura 2 muestra la dinámica de trabajo que desarrolla el alumno a lo largo del curso en el sistema adaptativo que se pretende mejorar con la metodología propuesta. En este tipo de sistemas, el alumno, al iniciar cada tema del curso, realizará una o varias actividades orientadas a determinar su nivel inicial de conocimiento (en nuestro caso, un test adaptativo sobre aspectos generales de la materia objeto de estudio). Establecido este nivel de partida, el sistema va presentando al alumno actividades de muy distinta naturaleza, del grado de dificultad que considera adecuado para su nivel. Durante la

realización de dichas actividades, el sistema registra información relacionada con su desarrollo. Al final de cada tema, se suelen presentar unas actividades de evaluación, utilizadas para recalcular el nivel al que se encuentra el alumno, y determinar si puede pasar a los siguientes temas o debe repetir el mismo tema incrementando el grado de dificultad.

En la Figura 3 se muestra un mismo tema del curso de Linux en dos niveles de dificultad distintos (principiante y medio). Se puede apreciar como además de ser distinto el color de fondo de las páginas web, los conceptos que forman los temas también son distintos (menú de la parte izquierda en la Figura 3).

El sistema AHA va almacenando la información de utilización de los usuarios en ficheros logs que posteriormente hemos trasladado a una base de datos. Este paso de ficheros de texto a base de datos se realiza para facilitar y aumentar la velocidad del algoritmo de descubrimiento de reglas, ya que debido a la gran cantidad de información es mucho más lógico trabajar con una base de datos. Durante este traspaso de información también se ha realizado un preprocesado de los datos, consistiendo principalmente en la eliminación de la información no útil, en la construcción de los atributos y en la discretización de los valores del atributo tiempo. Finalmente los datos de utilización que vamos a utilizar son relativos a: los tiempos de visualización de las páginas, aciertos obtenidos en las preguntas y niveles de conocimiento. De forma que, para cada estudiante, el sistema tiene información sobre los siguientes aspectos:

- **Tiempos.** Son los tiempos empleados en la realización de las preguntas, de las actividades y en la visualización de los contenidos de cada página web del curso.
- **Aciertos.** Son los aciertos o fallos cometidos en las distintas preguntas realizadas en el curso. Tanto los test como las actividades está formadas por una serie de preguntas.
- **Niveles.** Son los niveles obtenidos en las distintas actividades y test que componen el curso. Se han distinguido 3 niveles: principiante, medio y avanzado. Para asignarlos se utiliza un test adaptativo al comienzo del tema, una actividad por cada concepto que compone el tema y un test clásico al final del tema.

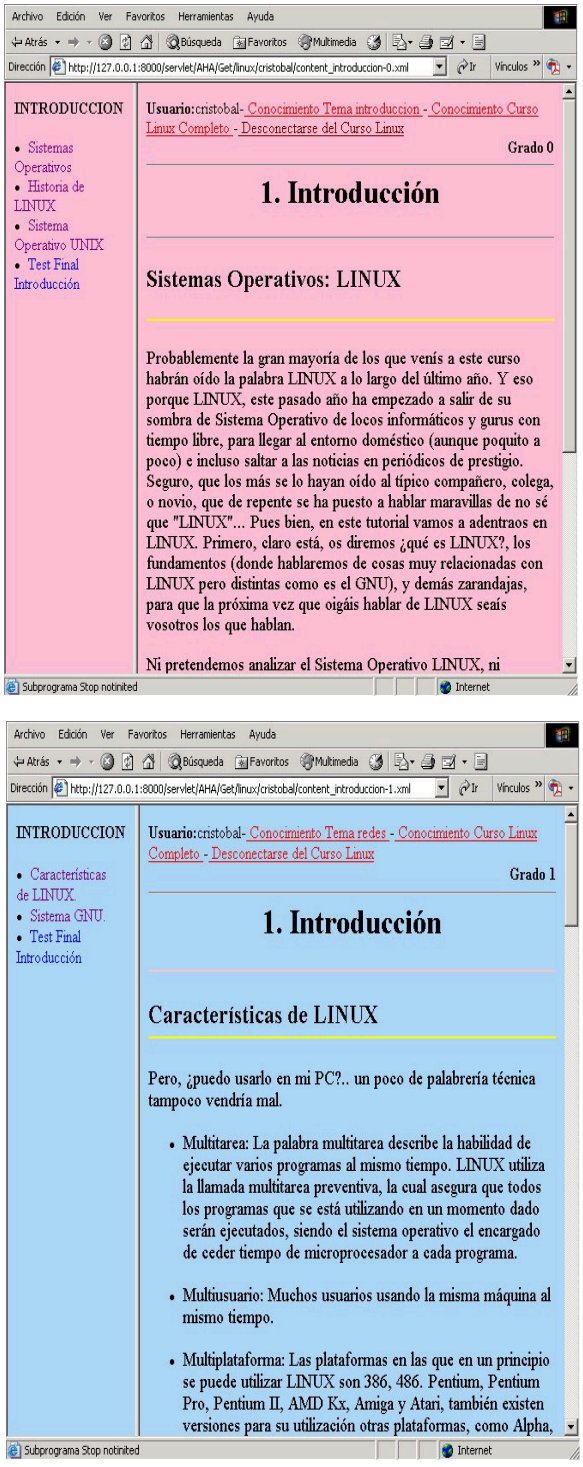


Figura 3. Dos niveles de dificultades distintas del mismo capítulo de introducción del curso de Linux.

6. Descubrimiento de Reglas de Predicción utilizando Algoritmos Evolutivos

La tarea del descubrimiento de reglas ha sido abordada desde multitud de paradigmas: construcción de árboles de decisión, aprendizaje inductivo, aprendizaje basado en instancias y, más recientemente redes neuronales y algoritmos evolutivos [Freitas 02]. El tipo de búsqueda que realizan cada uno de estos algoritmos va a determinar dónde se encuentran localizados dentro del panorama de la minería de reglas y desde el punto de vista de la minuciosidad de la búsqueda (ver Figura 4).

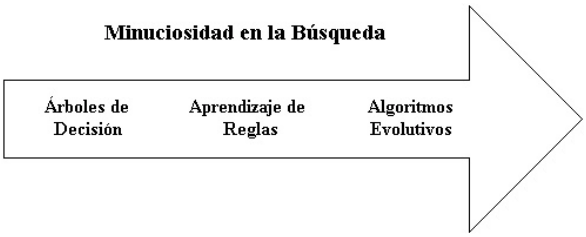


Figura 4. Minuciosidad de búsqueda de los algoritmos de descubrimiento de reglas.

En la Figura 4 se muestra el espectro de las técnicas de búsqueda en términos de la minuciosidad de la búsqueda que realizan. Por un lado del espectro están los algoritmos de inducción de reglas mediante árboles de decisión que utilizan heurísticas altamente voraces y realizan una búsqueda irrevocable. Los algoritmos de inducción de árboles son actualmente las técnicas más utilizadas en minería de datos. Son muy rápidos y sorprendentemente efectivos para encontrar clasificadores precisos, además de clasificar completamente los datos. Pero los algoritmos de inducción de reglas generalmente pierden parte de exactitud por su velocidad. La mayoría utilizan técnicas de particionado recursivo que van partiendo el conjunto de datos utilizando heurísticas voraces que pueden pasar por alto relaciones multivariables que no aparecen si se tratan las variables individuales aisladamente. Justo al lado del espectro se encuentran los algoritmos de aprendizaje de reglas convencionales donde se consideran una amplia variedad de alternativas cuya característica común es la de ser más minuciosos que los anteriores. Y en el otro lado del espectro se encuentran los algoritmos evolutivos que son capaces de conducir muchas búsquedas minuciosas y realizar un retroceso implícito en la búsqueda del espacio de reglas que va a permitir encontrar interacciones complejas que los otros tipos de algoritmos no son capaces de encontrar.

Los Algoritmos Evolutivos son algoritmos estocásticos de búsqueda basados en las ideas de la evolución darwiniana. Los paradigmas de Computación Evolutiva que se han aplicado para resolver el problema del descubrimiento de reglas [Freitas 02] son los Algoritmos Genéticos y la Programación Genética. La Programación Genética se puede considerar como un paradigma de búsqueda más abierta que el de Algoritmos Genéticos. La búsqueda realizada por la GP puede ser muy útil para clasificación y otras tareas, ya que el sistema puede producir diferentes combinaciones de atributos, utilizando las funciones disponibles en un conjunto preestablecido por la codificación, que no se considerarían utilizando un algoritmo genético convencional. La Programación Genética basada en gramáticas [Whigham 95] es un paradigma de programación genética en el que los individuos vienen representados como árboles de derivación de una gramática definida por el usuario para especificar el espacio de soluciones al problema. Se ha elegido este paradigma por la expresividad que presenta, que va a facilitar enormemente la interacción con el usuario. A continuación vamos a describir el algoritmo evolutivo utilizado y la codificación y la función de evaluación concreta de los individuos.

6.1 Algoritmo Evolutivo

El algoritmo evolutivo que hemos empleado para realizar la tarea específica de descubrimiento de reglas de predicción [Romero et al. 04], se muestra en la Figura 5.

Figura 5. Algoritmo evolutivo utilizado para la búsqueda de reglas de predicción.

El algoritmo comienza con la inicialización de la población, consiste en la generación del conjunto o población inicial de individuos o reglas. A

continuación se realiza la evaluación, que consiste en calcular el ajuste de cada individuo y el almacenamiento de los mejores (en nuestro caso los no dominados) en una población final. Después se determina si el algoritmo debe de terminar, es decir, si ha alcanzado un número determinado de evoluciones o ha encontrado un número determinado de reglas. Sino, se pasa a la etapa de selección donde se elige de entre la población actual y la final a los individuos que van a ser padres de la siguiente etapa de reproducción. La reproducción consiste en la creación a partir de los padres seleccionados de nuevos individuos mediante los operadores de cruce y mutación [Michalewicz 96]. Finalmente la población actual es sustituida por la nueva población de padres y el proceso se vuelve a repetir.

6.2 Codificación de las Reglas

Hemos utilizado una aproximación tipo Michigan [Zytkow et al. 01] para la representación de los individuos, donde cada individuo representa una regla. Para la codificación de los individuos hemos utilizado una gramática que en forma de Backus Naur (BNF) es la siguiente.

```
<regla>::="SI" <antecedente> "ENTONCES" <consecuente>
<antecedente>::=<antecedente> "Y" <condición> | <condición>
<consecuente>::=<condición>
<condición>::=<atributonivel> "=" <valornivel> |
              <atributotiempo> "=" <valortiempo> |
              <atributoacierto> "=" <valoracierto>
<atributonivel>::="TIEMPO."Nombre de atributo nivel válido
<atributoacierto>::="ACIERTO."Nombre de atributo acierto válido
<atributotiempo>::="NIVEL."Nombre de atributo tiempo válido
<valornivel>::="PRINCIPIANTE" | "MEDIO" | "EXPERTO"
<valoracierto>::="SI" | "NO"
<valortiempo>::="ALTO" | "MEDIO" | "BAJO"
```

Como puede comprobarse, los individuos generados a partir de esta gramática son reglas de predicción, en las que el antecedente puede presentar una o más condiciones y el consecuente presenta una única condición. Cada una de las posibles condiciones relaciona un atributo de la tabla con uno de los valores posibles que presenta el atributo.

6.3 Función de Evaluación

La función de evaluación mide la calidad de los individuos o reglas en nuestro caso. En la bibliografía hay descritas una gran cantidad de métricas para evaluar reglas [Lavrac et al. 99] [Tan et al. 00] (soporte, confianza, interés, precisión, informatividad, fiabilidad negativa, sensibilidad, especificidad, cobertura, innovación, satisfacción, precisión relativa, etc.). Pero cada una mide un aspecto de la regla. Este problema sugiere el uso de una aproximación multiobjetivo [Fonseca et al. 93] para el descubrimiento de reglas, donde el valor de la función ajuste a optimizar no es un valor escalar único, sino un vector de valores, donde cada valor mide un aspecto diferente de la calidad de la regla. En nuestro caso la función de evaluación o función de ajuste utilizada esta formada por un vector de tres componentes donde cada uno mida uno de los siguientes criterios de los individuos:

- **Exactitud de la regla.** Mide la exactitud o precisión de las reglas. Nosotros hemos utilizado la medida denominada factor de certeza [Shortliffe et al. 75].
- **Comprensibilidad de la regla.** Mide la comprensibilidad de la regla por parte del usuario. Nosotros hemos utilizado la medida de simplicidad [Liu et al. 00] que depende de la longitud de la regla.
- **Interesabilidad de la regla.** Mide el interés objetivo y subjetivo de la regla. Nosotros hemos utilizado la medida de interés [Tan et al. 00].

7. Implementación

Para facilitar al profesor o autor del curso la realización del proceso de descubrimiento de reglas de predicción hemos desarrollado una herramienta específica de minería de datos denominada EPRules (Education Prection Rules), para que no tenga que utilizar alguna de las muchas herramientas de propósito general existentes como Weka [Witten et al. 99], DBMiner [Zytkow et al. 01] etc. EPRules se ha desarrollada en lenguaje Java y su principal característica es su especialización en educación, utilizando atributos concretos, filtros y restricciones específicas para datos de utilización de los cursos, por lo que se adapta mejor a entornos educativos que las herramientas de propósito general. Las principales tareas que se pueden realizar con ellas son:

- **Datos de entrada**. Desde esta ventana (ver Figura 6) se puede o bien abrir una base de datos ya existente con datos de utilización de un curso o bien crear una nueva, y añadirle nuevos alumnos. Para crearla o añadir datos se deben seleccionar los ficheros de utilización del curso

(ficheros logs del alumno) que serán preprocesados e integrados en una base de datos relacional. También se pueden seleccionar y configurar los parámetros del algoritmo de discretización de la variable tiempo.

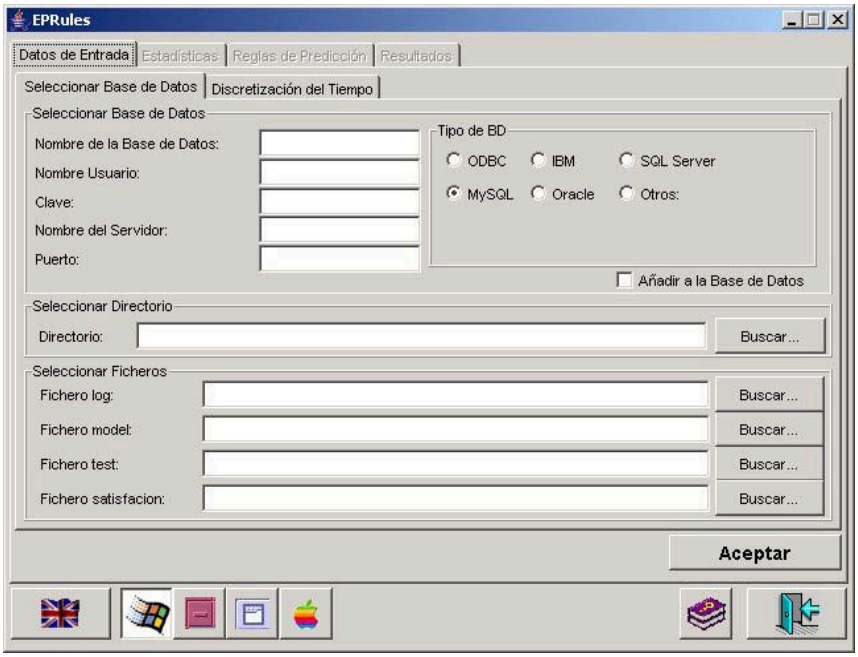
- 

Figura 6. Ventana de datos de entrada de EPRules

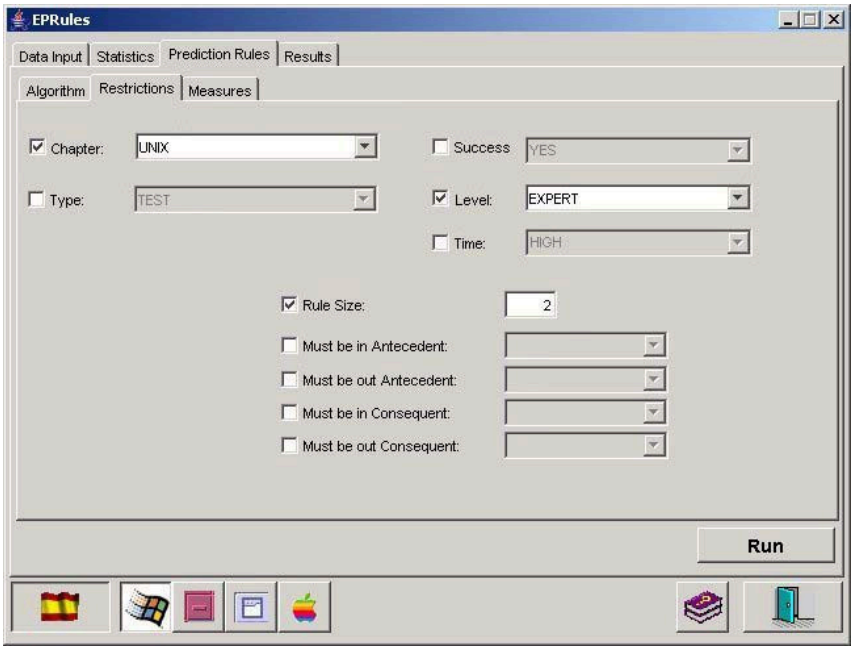
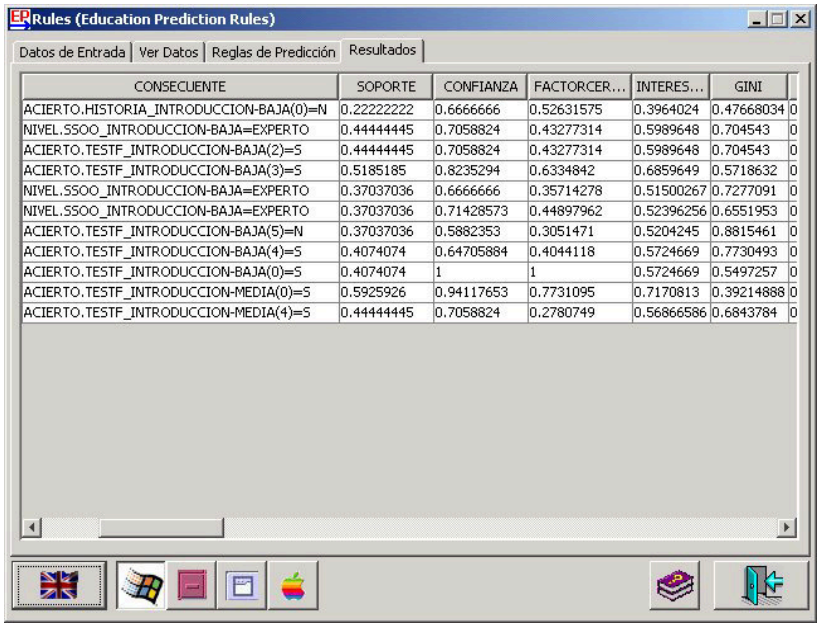


Figura 7. Ventana de selección de restricciones de EPRules.

- **Ver datos y estadísticas**. Desde esta ventana se puede visualizar todos los datos de utilización de los alumnos del curso ya preprocesados. Estos datos son sobre los tiempos, aciertos o niveles obtenidos por los alumnos en las distintas páginas web (actividades y contenidos) que componen el curso. Se pueden seleccionar desde visualizar los datos de todos los alumnos, hasta las de un alumno en concreto, o sólo para un tema determinado del curso, o sobre un concepto determinado de un tema, o a un nivel de visibilidad o dificultad de un tema determinado (Alto, medio, bajo) o sólo de un tipo de información determinada (tiempo, acierto o nivel).
- **Reglas de predicción.** Esta es la parte más importante de la herramienta ya que es desde donde se aplican los distintos algoritmos de descubrimiento de reglas disponibles. En principio los algoritmos que se han implementado son: un algoritmo de construcción de árboles de decisión como es el ID3 [Quilan 87], un algoritmo de reglas de asociación como es el Apriori [Agrawal et al. 94], un algoritmo de inducción de reglas como es el Prism [Cendrowska 87] y diferentes versiones de algoritmos evolutivos, en concreto de programación genética basada en gramática [Whigham 95] con o sin multiobjetivo [Fonseca et al. 93]. Se puede seleccionar tanto el algoritmo que se desea utilizar y sus parámetros de ejecución específicos como las restricciones subjetivas que deben de cumplir las reglas (ver Figura 7).
- **Resultados.** Esta ventana aparece automáticamente después de finalizar la ejecución un algoritmo y permite visualizar todas las reglas de predicción descubiertas (ver Figura 8). En concreto, para cada regla de predicción descubierta se muestra primero las condiciones que componen el antecedente y el consecuente de la regla, y después todos los valores para cada una de las medidas de evaluación [Tan et al. 00] [Lavrac et al. 99] de reglas (confidence, support, interest, gini, laplace, etc. hasta un total de 40 actualmente disponibles). Por defecto se muestran en el orden en el que se han ido descubriendo, pero se pueden reordenar por una condición o por el valor de cualquiera de las medidas, con sólo pulsar en la columna deseada.



CONSECUENTE	SOPORTE	CONFIANZA	FACTORCER...	INTERES...	GINI	
ACIERTO.HISTORIA_INTRODUCCION-BAJA(0)=N	0.22222222	0.6666666	0.52631575	0.3964024	0.47668034	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.44444445	0.7058824	0.43277314	0.5989648	0.704543	0
ACIERTO.TESTF_INTRODUCCION-BAJA(2)=S	0.44444445	0.7058824	0.43277314	0.5989648	0.704543	0
ACIERTO.TESTF_INTRODUCCION-BAJA(3)=S	0.5185185	0.8235294	0.6334842	0.6859649	0.5718632	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.37037036	0.6666666	0.35714278	0.51500267	0.7277091	0
NIVEL.SSOO_INTRODUCCION-BAJA=EXPERTO	0.37037036	0.71428573	0.44897962	0.52396256	0.6551953	0
ACIERTO.TESTF_INTRODUCCION-BAJA(5)=N	0.37037036	0.5882353	0.3051471	0.5204245	0.8815461	0
ACIERTO.TESTF_INTRODUCCION-BAJA(4)=S	0.4074074	0.64705884	0.4044118	0.5724669	0.7730493	0
ACIERTO.TESTF_INTRODUCCION-BAJA(0)=S	0.4074074	1	1	0.5724669	0.5497257	0
ACIERTO.TESTF_INTRODUCCION-MEDIA(0)=S	0.5925926	0.94117653	0.7731095	0.7170813	0.39214888	0
ACIERTO.TESTF_INTRODUCCION-MEDIA(4)=S	0.44444445	0.7058824	0.2780749	0.56866586	0.6843784	0

Figura 8. Ventana de resultados de EPRules.

8. Pruebas y Resultados Obtenidos

Para la realización de las pruebas hemos usado los datos de utilización del curso hipermedia adaptativo basado en web sobre el S.O. Linux que ha sido ejecutado por 50 alumnos del Ciclo Formativo Superior de Informática en el Instituto Gran Capitán de Córdoba y 10 estudiantes de Ingeniería Técnica Informática también de Córdoba.

Algoritmo	Número de reglas descubiertas		
	Todos	Rango	Frecuentes
ID3	474	131	89
Prism	657	172	62
Apriori	7960	491	70
AE-GBGP	198	162	51

Tabla 1. Comparación del número de reglas descubiertas.

Algoritmo	Porcentaje de reglas exactas		
	Todos	Rango	Frecuentes
ID3	46,0	51,9	60,3
Prism	71,9	53,7	91,9
Apriori	84,3	90,0	93,0
AE-GBGP	76,5	86,1	96,3

Tabla 2. Comparación del porcentaje de reglas exactas descubiertas.

Algoritmo	Porcentaje de reglas interesantes		
	Todos	Rango	Frecuentes
ID3	1,5	7,6	15,6
Prism	2,5	11,6	49,3
Apriori	3,6	7,9	53,1
AE-GBGP	21,9	60,4	76,6

Tabla 3. Comparación del porcentaje de reglas interesantes descubiertas.

Para probar la efectividad del algoritmo evolutivo primero lo hemos comparado con tres algoritmos clásicos en la búsqueda de reglas. Un algoritmo de descubrimiento de reglas de asociación como es el algoritmo Apriori [Agrawal et al. 94], otro de reglas de clasificación como es el ID3 [Quilan 87] y uno de inducción de reglas como el Prism [Cendrowska 87]. Pero los hemos tenido que modificar para que encuentren reglas con un solo consecuente y que tengan el mismo formato que nuestras reglas de predicción. Hemos utilizado de parámetros un tamaño máximo de regla de 3 y en el A priori un valor mínimo de soporte y confianza de 0.85. Y para el algoritmo evolutivo una población de 100 individuos de tamaño máximo 3, con una probabilidad de cruce de 0.8 y de mutación de 0.1, como valor mínimo de la medida de evaluación un valor de 0.85 y finalización en 50 evoluciones.

Los resultados obtenidos muestran (Ver Tabla 1, 2 y 3) que, en general, los algoritmos evolutivos generan un menor número de reglas y de mayor interés que los algoritmos clásicos, siendo más aptos para su utilización on-line como métodos de extracción de conocimiento en el sistema de enseñanza adaptativo. Los algoritmos clásicos, y sobre todo el Apriori producen, en general, reglas bastante exactas, pero fallan a la hora de generar reglas con interés elevado y, además, la longitud de las reglas que producen dificulta su comprensibilidad. Además, cuando el conjunto de partida es elevado (lo cual puede suceder cuando el usuario desea extraer información global acerca del sistema, sin aplicar ningún tipo de restricción sobre dicho conjunto), los generan un conjunto tan enorme de reglas que hace impide su aprovechamiento posterior. Los resultados obtenidos para los algoritmos evolutivos propuestos muestran que, en general, producen un menor número de reglas que los algoritmos clásicos, siendo esta diferencia de un orden de magnitud en los casos más favorables. Además, la proporción de reglas comprensibles e interesantes es superior. Además el uso de algoritmos basados en el concepto de frente de Pareto (MOGA y NSGA) [Fonseca et al. 93] permite optimizar los tres objetivos planteados de forma simultánea, produciendo en todas las ejecuciones la mayor proporción de reglas exactas, comprensibles e interesantes.

8. Descripción y Utilización de la Información Descubierta

El objetivo final como ya hemos comentado anteriormente es mostrar un conjunto de reglas interesantes al profesor para que pueda tomar decisiones sobre como mejorar el curso. Las reglas descubiertas presentan distintos tipos de relaciones dependiendo del tipo de los atributos del consecuente y antecedente:

- **Tiempo**. Muestra atributos (los del antecedente) que influyen en el tiempo (el del consecuente).
- **Nivel**. Muestra atributos (los del antecedente) que influyen en el nivel (el del consecuente).
- **Acierto**. Muestra atributos (los del antecedente) que influyen en el acierto (el del consecuente).

Estas relaciones además pueden hacer referencia a temas (niveles obtenidos en test iniciales y finales) o a conceptos (tiempos, aciertos y niveles obtenidos en páginas de contenido teórico y actividades de evaluación).

A partir de estas reglas el profesor puede realizar los cambios que desee oportunos en el curso para fortalecer la relación (si la considera deseable) o por el contrario eliminar la relación (si la considera como no deseable) cambiando los contenidos, la estructura y la adaptación del curso. A continuación se van a describir un par de reglas descubiertas.

Si *TIEMPO.TESTF_ADMINISTRACION-ALTA(0) = ALTO*
Entonces
ACIERTO. TESTF_ADMINISTRACION-ALTA(0) = NO
(Interés=0.51, Factor Certeza=0.79, Simpleza= 1)

Esta regla muestra la relación entre el tiempo empleado en leer una pregunta (pregunta número 0 del test final de grado ALTO del tema Administración) y el fallo en la contestación de dicha pregunta. Esta relación está confirmando que la pregunta no está bien formulada o tiene algún tipo de error, ya que no sólo se obtiene un tiempo alto en leerla, sino que además y de forma simultánea se responde incorrectamente. El diseñador ante este tipo de relaciones debe de corregir estas preguntas, modificando el enunciado si se encuentra el posible error o cambiándola por otra pregunta distinta. En este caso particular se comprobó que el test correspondiente del concepto ADMINISTRACION era confuso y no estaba bien formulado, y se paso a cambiarlo por otra pregunta de similares características, mejor definida.

Si *NIVEL.EMULADORES_PROGRAMAS-ALTA = EXPERTO*
Entonces
ACIERTO.EMULADORES_PROGRAMAS-ALTA(1) = NO
(Interés= 0.69, Factor Certeza= 0.73, Simpleza = 1)

Esta regla muestra la relación que existe entre el nivel final obtenido por el alumno en la actividad de evaluación de un concepto (concepto EMULADORES que tiene dificultad ALTA dentro del tema PROGRAMAS) y el fallo a una determinada pregunta de dicha actividad. Esta relación indica que una pregunta que no han acertado un número importante de alumnos de nivel EXPERTO puede estar mal planteada o no se entiende, pudiendo crear confusión en el alumno. En este caso particular se comprobó que el ítem EMULADORES_PROGRAMAS(1) era confuso en el planteamiento de la pregunta y se corrigió el problema.

Si *NIVEL.INTERFAZ_REDES-ALTA = EXPERTO*
Entonces
NIVEL.TCPIP_TELNET-MEDIA = EXPERTO
(Interés= 0.57, Factor Certeza= 0.75, Simpleza = 1)

Esta regla muestra que los niveles obtenidos en las actividades han sido simultáneamente altos. Esto indica que los conceptos asociados están relacionados. En este caso, el diseñador del curso debería comprobar el contenido de ambos conceptos para ver a qué se debe la relación y optar por: unir ambos conceptos en un único concepto, colocar ambos conceptos en una misma lección, asignarles el mismo grado de dificultad, corregir las reglas de asignación de niveles, etc. En este caso particular se consideró que el ambos conceptos debían tener el mismo grado de dificultad y estar unidos en un mismo concepto. Si los niveles se refieren a test, ya sea iniciales o finales en lugar de actividades, podemos concluir que los temas están relacionados. El diseñador del curso, en este caso, puede unir los temas, o ponerlos uno a continuación del otro.

Finalmente indicar que la mayoría de las reglas que realmente nos interesan por su utilidad, observamos que presentan un soporte bajo y una alta confianza. Significando que son reglas que no cumplen la gran mayoría de los alumnos, pero si un grupo más o menos reducido de alumnos con una alta fiabilidad, es decir, que si lo cumple una persona del grupo es seguro que lo cumple otra persona del grupo. Ya hemos visto como utilizando de referencia estas relaciones descubiertas el profesor puede decidir realizar los cambios que crea oportunos en el contenido, estructura o control del curso para potenciar o eliminar dichas relaciones según sean deseables o no deseables. Como por ejemplo: cambiando preguntas, actividades, páginas, reestructurando las relaciones de dependencia o requisitos entre conceptos, etc.

9. Conclusiones y Trabajo Futuro

En este trabajo se ha presentado una metodología para la mejora de sistemas hipermedia adaptativos educativos basados en web mediante el descubrimiento de reglas de predicción con algoritmos evolutivos. En concreto, se propuesto la utilización de programación genética basada en gramática con multiobjetivo. Los resultados, en función del número de reglas obtenidas y el grado de interés, precisión y comprensibilidad de las reglas, son en la mayoría de los casos superiores en comparación con los otros algoritmos clásicos propuestos, que utilizan una única medida o una composición de varias de evaluación de las reglas. Con respecto a la utilidad práctica de las reglas descubiertas para la toma de decisiones sobre posibles modificaciones que se pueden realizar en los cursos, se han descrito los distintos tipos de reglas, se han descrito las utilidades que pueden tener para la mejora del curso y se han mostrado ejemplos concretos de reglas descubiertas con el curso de Linux. Para facilitar la realización de todo este proceso de descubrimiento de conocimiento se ha desarrollado la herramienta específica EPRules que permite realizar el preprocesado de los datos de utilización de los cursos web, el establecimiento de restricciones sobre el tipo de información que se desea descubrir, así como la aplicación de los algoritmos de minería de datos para extracción de reglas y la visualización de las mismas. Actualmente estamos trabajando en la automatización completa del proceso de descubrimiento de conocimiento, de forma que las reglas descubiertas se puedan aplicar directamente sobre el curso, sin la necesidad de que el profesor o autor del curso tenga que realizarlas manualmente, sino solamente tenga que aceptar o rechazar la realización de los cambios propuestos por las reglas. Como línea futura de investigación se considera interesante la búsqueda de métricas relacionadas con el interés subjetivo que muestran los profesionales por las reglas descubiertas. En este sentido, existen referencias de algoritmos evolutivos [Williams 99] en los que no existe una función de aptitud, sino que los individuos son valorados por un experto en cada ciclo del algoritmo.

Referencias

[Agrawal et al. 94] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules" Proc. 20th Int. Conference on Very Large Databases. Santiago de Chile. 1994.

[Brusilovsky 98] P. Brusilovsky, "Adaptative Educational Systems on the World-Wide-Web: A Review" Int. Conf. on Intelligent Tutoring Systems. San Antonio. 1998.

[Cendrowska 87] J. Cendrowska "PRISM: an algorithm for inducing modular rules" Journal of Man-Machine Studies. 27, pp. 349-370. 1987.

[De Bra et al. 00] P. De Bra, H. Wu, A. Aerst, G. Houben, "Adaptation Control in Adaptive Hypermedia Systems" Int. Conf. on Adaptive Hypermedia. Trento, Italia. 2000.

[Fonseca et al. 93] C.M. Fonseca, P.J. Fleming, "Genetic Algorithms for Multiobjective Optimization" Conf. on Genetic Algorithms. San Mateo, CA. 1993.

[Freitas 00] A.A. Freitas, "Understanding the Crucial Differences Between Classification and Discovery of Association Rules" ACM SIGKDD Explorations, 2(1), 2000.

[Freitas 02] A.A. Freitas, "Data Mining and Knowledge Discovery with Evolutionary Algorithms" Springer-Verlag. 2002.

[Hipp et al. 00] J. Hipp, U. Güntzer, G. Nakhaeizaeh, "Algorithms for Association Rule Mining. A General Survey and Comparison" ACM SIGKDD. 2000.

[Lavrac et al. 99] N. Lavrac, P. Flach, B. Zupan, "Rule Evaluation Measures: A Unifying View" ILP-99, LNAI 1634. Springer-Verlag Berlin Heidelberg. 1999.

[Liu et al. 00] J.L. Liu, J.T. Kwok "An Extended Genetic Rule Induction" Conf. Evolutionary Computation. 2000.

[Michalewicz 96] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs" 3rd edn. Springer-Verlag, Berlin Heidelberg New York. 1996.

[Minaei-Bidgoli et al. 03] B. Minaei-Bidgoli, W.F. Punch "Predicting student performance: an

application of data mining methods with the educational web-based system LON-CAPA“ IEEE Frontiers in Education. Pp 1-6. 2003.

[Quilan 87] J.R. Quilan “Generating Production rules from decision trees” Proceeding of IJCAI-87. 1987.

[Romero et al. 02] C. Romero, P. De Bra, S. Ventura, C. De Castro “Using Knowledge Level with AHA! For Discovering Interesting Relationship” ELEARN. Montreal. 2002.

[Romero et al. 04] C. Romero, S. Ventura, P. de Bra “Knowledge Discovery with Genetic Programming for Providing Feedback to Courseware Author“ User Modeling and User-Adapted Interaction. Vol. 14. No. 5. pp. 425-464. 2005.

[Scime 04] A. Scime “Web Mining: Applications and Techniques” Idea Group. 2004.

[Shortliffe et al. 75] E. Shortliffe, B. Buchanan “A model of inexact reasoning in medicine” Mathematical Biosciences, 23 pp. 351-379. 1975.

[Srivastava et al. 00] J. Srivastava, B. Mobasher, R. Cooley “Automatic Personalization Based on Web Usage Mining” Communications of the Association of Computing Machinery. pp. 142-151. 2000.

[Tan et al. 00] P. Tan, V. Kumar “Interesting Measures for Association Patterns” Technical Report TR00-036. Department of Computer Science. University of Minnnesota. 2000.

[Wang 02] F. Wang “On Analysis and Modeling of Student Browsing Behavior in Web-Based Asynchronous Learning Environments“ Int. Conf. on Web-based Learning. pp. 69-80. 2002.

[Whigham 95] P.A. Whigham “Gramatically-based Genetic Programing” Proceeding of the Workshop on Genetic Programming. pp. 33-41. 1995.

[Williams 99] G.J. Williams “ Evolutionary Hot Spots Data Mining. An Architecture for Exploring for Interesting Discoveries” Conf on Knowledfe Discovery and Data Mining. 1999.

[Witten et al. 99] I.H. Witten, E. Frank “Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations” Morgan Kaufmann. 1999.

[Yu et al. 01] P. Yu, C. Own, L. Lin “On the Learning Behavior Analysis of Web Based Interactive Environmnent” ICCE. 2001.

[Zaïene 01] O. R. Zaïene, “Web Usage Mining for a Better Web-Based Learning Environment” Conf. on Advanced Technology for Education. Alberta. 2001.

[Zaïane 02] O.R. Zaïane “Building a Recommender Agent for e-Learning Systems” International Conference on Computers in Education. New Zealand. pp 55-59. 2002.

[Zytkow et al. 01] J. Zytkow, W. Klosgen “Handbook of Data Mining and Knowledge Discovery” Oxford University Press. 2001.