APLICACIÓN DE ALGORITMOS EVOLUTIVOS COMO TÉCNICA DE MINERÍA DE DATOS PARA LA MEJORA DE CURSOS HIPERMEDIA ADAPTATIVOS BASADOS EN WEB

(Applying Evolutionary Algorithms as Data Mining Methods to Improve Web-based Adaptive Hypermedia Courses)

Critóbal Romero Morales
Sebastián Ventura Soto
Carlos de Castro

Departamento de Informática y Análisis Numérico
Universidad de Córdoba. (España)

Resumen: Este artículo ilustra el uso de algoritmos evolutivos como técnica de minería de datos para el descubrimiento de reglas de predicción en bases de datos, reglas que se utilizarán en la mejora de Cursos Hipermedia Adaptativos basados en Web. La idea consiste en descubrir relaciones importantes entre los datos de utilización recogidos durante las ejecuciones de los distintos usuarios. Esta información puede ser de gran utilidad para el creador del curso, que puede decidir qué modificaciones son las más adecuadas para mejorar el rendimiento de los alumnos. Para la realización de la búsqueda de reglas de predicción se ha utilizado Programación Genética Basada en Gramáticas (GBGP) con técnicas de optimización multiobjetivo. El trabajo también presenta la herramienta gráfica de descubrimiento de reglas que se ha desarrollado para facilitar la utilización de la metodología propuesta de mejora de cursos.

Minería de Datos – Cursos Web Adaptativos – Algoritmos Evolutivos – Reglas de Predicción.

Abstract: This paper shows how to use evolutionary algorithms to data mining methods for discovering prediction rules in databases. These rules will be used to improve web-based adpative hypermedia courses. The idea is to discover important rules among the usage data picked up during the students' executions. This information may be very useful to the author of the course, who can decide what modification will be the most apropiate to improve the performance of

students. In order to do the discovering of rules we have used grammar-based genetic programming (GBGP) with multiobjective optimization technics. In this work we also present a graphic tools for discovering rules to facilitate the usage of the proposed methodology to improve the courses.

Data Mining – Web-based Adaptive Courses – Evolutionary Algorithms – Prediction Rules.

1. INTRODUCCIÓN

El desarrollo de cursos hipermedia adaptativos basados en web es una actividad laboriosa (Hérin et al., 2002), tanto más compleja cuanto mayor es el número de posibilidades de adaptación que se desea ofrecer. Por ejemplo, se deben elegir los contenidos que serán mostrados y establecer la estructura del curso, determinar cuáles de estos contenidos son los más adecuados para cada usuario potencial y cuál es la mejor organización de estos contenidos, aprovechando las posibilidades de adaptación ofrecidas por el sistema. Sin embargo, un diseño cuidadoso no suele ser suficiente sino que, en la mayoría de los casos, suele ser necesario realizar una evaluación posterior basada en los resultados obtenidos por los usuarios del mismo. Sería además deseable que esta actividad de autoevaluación se realizara de forma continua mientras el curso se encuentre a disposición de los alumnos (Ortigosa & Carro, 2002). Para ello es necesario utilizar herramientas y metodologías capaces de observar el comportamiento de los estudiantes y de asistir al profesor en el proceso de mejora continua de los cursos adaptativos, detectando de forma semiautomática posibles errores, carencias o mejoras que puedan realizarse en los cursos ya generados.

Para abordar el problema planteado, se están comenzando a utilizar técnicas de descubrimiento de conocimiento o minería de datos (Zaïne, 2001) que asisten al profesor en la validación de los cursos. Estas técnicas permiten descubrir nuevo conocimiento a partir de los datos de utilización del curso por parte de los alumnos. La idea se ha aplicado ya con éxito en los sistemas de comercio electrónico[1], donde ha llegado a ser muy popular (Spiliopoulou, 2000). Sin embargo, se ha hecho muy poco en este sentido para la comprensión del comportamiento de los alumnos en los entornos de aprendizaje a distancia basados en web.

La minería de datos (Data Mining, DM) es un área multidisciplinar, donde confluyen multitud de paradigmas de cómputo tales como: la construcción de

[1] Por ejemplo, se han desarrollado herramientas para la comprensión del comportamiento de los clientes y ofrecerle productos relacionados con sus hábitos de compra para aumentar las ventas.

árboles de decisión, la inducción de reglas, las redes neuronales artificiales, el aprendizaje basado en instancias, el aprendizaje bayesiano, la programación lógica y varios tipos de algoritmos estadísticos (Witten & Frank, 2000). El objetivo de todos estos paradigmas es descubrir conocimiento nuevo, útil e interesante. En este sentido, desde hace relativamente poco tiempo se ha planteado la posibilidad de utilizar Algoritmos Evolutivos para resolver esta tarea (Freitas, 2002). Las principales ventajas del uso de los Algoritmos Evolutivos en el descubrimiento concreto de reglas de predicción es su capacidad para realizar una búsqueda global y el tratamiento óptimo del problema de la interacción de los atributos que la mayoría de los algoritmos voraces de inducción de reglas que se han utilizado tradicionalmente.

En el contexto de descubrimiento de reglas de predicción utilizando Algoritmos Evolutivos, un individuo corresponderá a una regla o conjunto de reglas candidatas; la función de ajuste corresponderá a alguna medida de la calidad de la regla o conjunto de reglas; el procedimiento de selección utilizará los valores del ajuste de los individuos para seleccionar las mejores reglas o conjunto de reglas; los operadores genéticos transformarán una regla candidata en otra. Los algoritmos evolutivos realizarán una búsqueda en el espacio de reglas candidatas como haría un método de inducción de reglas. La principal diferencia entre los algoritmos evolutivos empleados para el descubrimiento de conocimiento y los algoritmos de inducción de reglas es la estrategia de búsqueda empleada. En efecto, los algoritmos clásicos de aprendizaje inductivo suelen utilizar una estrategia voraz de búsqueda local, mientras que los algoritmos evolutivos utilizan una estrategia de búsqueda global inspirada en la selección natural.

Siempre es deseable que el conocimiento descubierto sea interesante (Bayardo & Agrawal, 1999), entendiendo como tal que sea interesante, novedoso y pueda ser aprovechado por el usuario. Para satisfacer estas exigencias, se deben extraer sólo un subconjunto de reglas interesantes de entre todas las posibles reglas descubiertas y, aunque los algoritmos de minería de datos normalmente descubren una gran cantidad de reglas exactas y comprensibles, no suelen ser reglas interesantes al ser éste un objetivo más ambicioso y difícil. Existen un gran número de métricas descritas en la bibliografía (Tan & Kumar, 2000), cada una de las cuáles se centra en un aspecto concreto de la calidad de las reglas descubiertas. Sin embargo, para la tarea que nos ocupa, no se ha encontrado una medida que conduzca a un rendimiento significativamente mejor que las demás. Por esta razón, se hace necesario considerar varias medidas simultáneamente. Una forma simple de hacer esto es utilizar una métrica combinada, que pondera mediante pesos algunas de las métricas primitivas (Noda et. al, 1999). Sin embargo, ésta no es una buena aproximación debido a que las medidas utilizadas pueden estar en conflicto y ser no conmensurables, en el sentido de que evalúan aspectos muy diferentes de la solución candidata. Este problema sugiere el uso de

una aproximación multiobjetivo (Freitas, 2002) 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. Aunque la bibliografía de Algoritmos Genéticos Multiobjetivo (MultiObjective Evolutionary Algorithms, MOEA) es amplia (Deb 2001), la utilización de MOEA en el descubrimiento de reglas parece relativamente inexplorada.

Es evidente que una condición para el éxito de nuestra propuesta de mejora de los cursos hipermedia adaptativos basados en web es la constatación de que los algoritmos de obtención de conocimiento van a aportar información relevante para la mejora del sistema, y ese es el objetivo fundamental que se persigue con este trabajo. Es, por tanto, fundamental disponer de un sistema que sirva como banco de pruebas para la obtención de información que se utilizará en la fase de extracción de conocimiento. Desgraciadamente, no se ha encontrado ningún sistema con todas las características requeridas para realizar esta fase de recogida de información. Por esta causa, se decidió elaborar uno propio aprovechando la arquitectura del sistema AHA (de Bra et al., 1999), una herramienta de dominio público[2] que presentaba mucha similitud con la herramienta que necesitábamos. Este sistema ya implementado sirvió como plataforma para el desarrollo de un curso sobre el Sistema Operativo Linux, que fue ejecutado por 40 alumnos del I.E.S. «Gran Capitán» de Córdoba y 10 expertos informáticos en dicho sistema operativo. La información generada por los alumnos del curso es la que se ha utilizado en la fase de extracción de conocimiento, que ha motivado el desarrollo de la investigación que se describe en este trabajo.

La estructura que vamos a seguir en el artículo es: Primero se va a describir la metodología propuesta para la mejora en el desarrollo de los cursos hipermedia adaptativos basados en web. Después se va a plantear el problema concreto de minería de datos que se desea resolver, en nuestro caso el descubrimiento de reglas de predicción. A continuación se va a describir la utilización de los algoritmos evolutivos para resolver esta tarea y el algoritmo que se ha implementado. También se va a describir la herramienta gráfica que va a facilitar la realización de todo el proceso de descubrimiento de conocimiento. Finalmente se muestran las principales conclusiones a las que se ha llegado tras realizar dicho trabajo.

[2] El sistema AHA se distribuye con licencia GPL, por lo que podía accederse a su código fuente y modificarlo en base a nuestros requisitos.

2. METODOLOGÍA PARA LA MEJORA DE LOS CURSOS

La metodología (Romero et al., 2002) que se propone para el desarrollo de los sistemas adaptativos para educación basada en Web (ver Figura 1) es una metodología cíclica, que consta de las siguientes etapas:

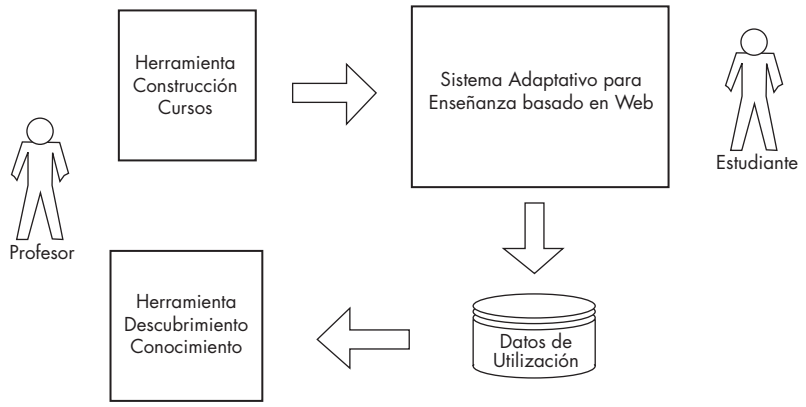


FIGURA 1. Metodología propuesta para la mejora de cursos.

- **Construcción del curso**. Es la primera etapa, y es donde se elabora el curso. Se corresponde con las fases de consenso del contenido del curso, recopilación de información sobre el tema, planificación del contenido y composición del curso, de la metodología clásica de construcción de cursos (Hérin et al. 2002). El diseñador del curso suele ser el responsable de construir el curso proporcionando para ello la información del modelo del dominio (contenido), el modelo pedagógico (reglas) y el módulo interfaz (aspecto gráfico). Normalmente suele utilizar una herramienta autor para facilitar esta tarea, ya sea una herramienta genérica de tipo comercial como DreamWeaver, Toolbook, Director, etc. o una herramienta específica diseñada para un tipo de curso concreto. El resto de información, modelo del tutor (motor de ejecución del sistema adaptativo) y modelo del alumno (información del estudiante) suelen venir dada por el propio sistema adaptativo de soporte de enseñaza utilizado. Para poder aplicar nuestra metodología, es necesario que el sistema desarrollado almacene la información relativa a la interacción del usuario con el sistema. Al finalizar esta etapa el curso debe colocarse en un servidor web para que los alumnos puedan utilizarlo de forma remota.

- Ejecución del curso. Es la realización del curso por parte de los alumnos. Los estudiantes, utilizando un navegador web, se conectarán al sitio que alberga la aplicación para realizar el curso. Durante esta etapa se recogerá la información de utilización de forma transparente para el alumno, almacenándose en el servidor web dentro de los distintos ficheros históricos o logs.
- Aplicación de técnicas de minería de datos. Consiste en la aplicación de técnicas de descubrimiento de conocimiento sobre los datos de utilización recogidos en la etapa anterior de ejecución del curso. Esta etapa se añadiría a la etapa de evaluación del curso de la metodología clásica. El objetivo es procesar toda la información almacenada en el sistema, previo preprocesado y colocación en un sistema de gestión de bases de datos que garantice una manipulación más rápida de dicha información. Una vez transferidos los datos, el diseñador del curso puede aplicar los algoritmos de minería de datos y descubrir relaciones importantes entre éstos.
- Mejora del curso. En esta etapa de mantenimiento se realizarán modificaciones sobre el curso orientadas a solventar carencias, corregir problemas o mejorar determinados aspectos del mismo. Esta etapa de mantenimiento está asistida por las técnicas de adquisición de conocimiento, diferenciándose en eso de la revisión que podría llevarse a cabo en algunos sistemas tradicionales. El diseñador del curso, ayudado por la información que se le suministra en forma de relaciones importantes descubiertas entre los datos, realiza las modificaciones que crea más adecuadas para mejorar el rendimiento del curso. Estas modificaciones pueden afectar al contenido del curso, su estructura, el interfaz gráfico, etc.

Como puede comprobarse, la introducción de una etapa de mantenimiento basada en el uso de técnicas de aprendizaje automático mejoraría sensiblemente la calidad del sistema, dada la posibilidad de aprovechar la enorme cantidad de información que se genera como consecuencia de la interacción de los alumnos con el sistema. Además, y dado que este ciclo puede repetirse cuantas veces se desee con un coste relativamente bajo (la fase de extracción de conocimiento se realiza automáticamente), se confiere un carácter dinámico al ciclo de vida de la aplicación, que puede ir mejorando progresivamente a medida que se dispone de más información.

3. DESCUBRIMIENTO DE REGLAS DE PREDICCIÓN

El uso de reglas es una de las formas más popular de representación del conocimiento debido, entre otras razones, a su sencillez, capacidad de expresión y escalabilidad. Dependiendo de la naturaleza del conocimiento que almacenan, se ha establecido una tipología informal para es tipo de estructuras. Así, se

habla de reglas de decisión, asociación, clasificación, predicción, causalidad, optimización etc. En el ámbito de la extracción de conocimiento en bases de datos, las más estudiadas han sido las reglas de asociación, de clasificación y de predicción.

La Figura 2 muestra el formato genérico de las reglas de predicción en formato EBNF. Como puede observarse, comparte características sintácticas con las reglas de asociación y con las de clasificación. Al igual que sucede en el caso de las reglas de clasificación, el consecuente sólo presenta una condición. Sin embargo, el contenido de este consecuente no tiene que ser un identificador de categoría o clase, sino que puede tratarse de una condición de cualquier tipo (que es la que se pretende predecir, de ahí el nombre de reglas de predicción). El significado semántico de una regla de predicción es que si todas las condiciones especificadas por el antecedente de la regla son satisfechas por los atributos predictores de un ejemplo, la regla predice que el atributo objetivo (el que aparece en el consecuente) de esa instancia tendrá el valor especificado en el consecuente de la regla.

```
Regla          ::= «SI» (<antecedente>) «ENTONCES» <consecuente>
<antecedente>::= <condición> | <condición> «Y» <antecedente>
<consecuente  ::= <condición>
<condición>    ::= <atributo> <operador> <valor> | <atributo> <operador> <atributo>
<atributo>     ::= Cada uno de los posibles atributos del conjunto
<valor>        ::= Un valor del dominio del atributo correspondiente
<operador>     ::= «=» | «<» | «>» | «» / «» / «»
```

FIGURA 2. Defición de las reglas de predicción en notación EBNF

Además de las diferencias sintácticas existentes entre este tipo de reglas, es importante recordar la diferencia en su significado, la cual condiciona de algún modo la forma de llevar a cabo la minería de un tipo de reglas determinado. En este sentido, Alex Freitas (Freitas, 2000) expone la naturaleza determinista que presenta la tarea de obtención de reglas de asociación, que establece relaciones entre atributos con unas garantías mínimas expresadas a través del soporte y confianza de la regla, frente a la naturaleza mal definida y no determinista que supone la tarea de extracción de reglas de clasificación que, en general, utiliza un conjunto de datos de entrenamiento para construir un clasificador que empleará sobre datos no pertenecientes a dicho conjunto.

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 (Witten & Frank, 2000). 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 3).

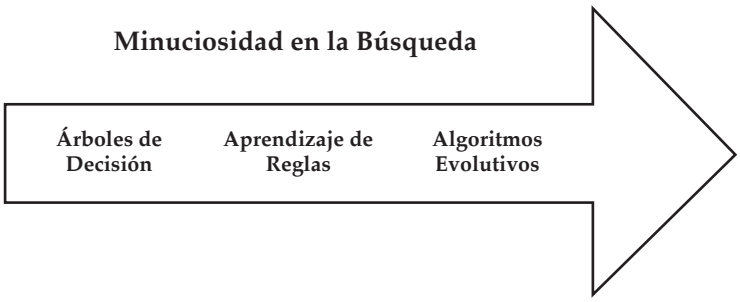


FIGURA 3. Minuciosidad de búsqueda de los algoritmos de descubrimiento de reglas.

En la Figura 3 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.

4. PROGRAMACIÓN GENÉTICA BASADA EN GRAMÁTICA PARA DESCUBRIMIENTO DE REGLAS DE PREDICCIÓN

Los Algoritmos Evolutivos (Evolutionary Algorithms, EA) 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 son los Algoritmos Genéticos (Genetic Algorithms, GA) y la Programación Genética (Genetic Programming, GP). 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 (Grammar Based Genetic Programming, GBGP) (Whigham, 1995) 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.

4.1 Algoritmo evolutivo

Se ha implementado un algoritmo evolutivo para el descubrimiento de reglas de predicción (ver Figura 4). Como puede comprobarse, el algoritmo utiliza dos conjuntos de individuos. El primero de ellos se corresponde con la población, mientras que el segundo almacena una élite de individuos que son los que serán devueltos al usuario tras finalizar el algoritmo. Tras la inicialización de la población, se eligen los padres a partir de la población actual (cuya diversidad está garantizada, como ser verá) y del conjunto élite (para garantizar que haya padres de calidad). Una vez generados los hijos tras la aplicación de operadores genéticos, se actualiza la élite añadiendo los individuos con mejores propiedades no existentes aún y, posteriormente, se actualiza la población, garantizando la diversidad de ésta. La evolución finaliza cuando se alcanza un máximo de generaciones.

```
Inicio
    Crear la población inicial P y el almacén élite E
    Mientras (generación < MaxGeneraciones) hacer
    Seleccionar los futuros padres a partir de los individuos de P y E
    Aplicar operadores genéticos sobre los padres seleccionados
    Evaluar los hijos obtenidos
    Actualizar los conjuntos P y E
    generación ¨ generación + 1
    Fin Mientras
    Devolver el conjunto E
Fin
```

FIGURA 4. Algoritmo Evolutivo empleado.

4.2. Representación de los individuos

La Figura 5 representa la gramática empleada para representar las reglas que van a ser buscadas por el algoritmo evolutivo. Como puede comprobarse, se trata de 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 relacionan un atributo de la tabla con uno de los valores posibles que presenta el atributo. Por razones de presentación[3], no se han incluido los símbolos terminales que representan las posibles etiquetas para cada uno de los atributos de la tabla (atributos relacionados con aciertos, tiempo y nivel).

```
<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"
```

FIGURA 5. Gramática de reglas de predicción en formato BNF.

[3] Hay que tener en cuenta que, en este tipo de problemas el número de posibles atributos es enorme.

Un posible árbol de derivación de esta gramática produciría la siguiente regla ejemplo:

SI ((ACIERTO.CARACTE_INTRODUCCION-ALTA(2) = NO) Y
(TIEMPO.CARACTE_INTRODUCCION-ALTA(2) = ALTO))
ENTONCES
NIVEL.CARACTE_INTRODUCCION-ALTA = EXPERTO

que está indicando que los alumnos que han sido finalmente evaluados como expertos en el concepto «Características del sistema Linux», no aciertan la pregunta número dos dentro de la actividad de evaluación de dicho concepto (que presenta un grado de dificultad elevado dentro del tema Introducción) y que además, invierten un tiempo elevado en la realización de esta pregunta.

4.3. Operadores genéticos

Los operadores genéticos son los encargados de generar los nuevos individuos a partir de los individuos de la población actual. Los operadores genéticos típicos en GBGP son los denominados por P. Whigham (Whighan, 1995) cruce selectivo y mutación selectiva. El operador de cruce selectivo es análogo al cruce de subárboles que se presenta en cualquiera de los paradigmas de programación genética. Sin embargo, por la estructura de los árboles de derivación el punto de cruce elige siempre a símbolos no terminales de la gramática. Para que la operación sea sintácticamente correcta, se obliga a que las raíces de los dos subárboles que se intercambian sean idénticas. La mutación selectiva reconstruye un subárbol tomando como raíz un nodo no terminal del mismo, sin más restricción que el número máximo de producciones que puede tener el árbol de derivación resultante.

4.4. Función de ajuste

El proceso de valoración de los individuos consiste en obtener una regla correcta a partir del árbol de derivación que contiene[4] y, posteriormente, aplicar una función que produzca una medida de la calidad de la regla. El conocimiento descubierto por un algoritmo de minería de datos debe satisfacer tres criterios principales (Freitas, 2002): exactitud, comprensibilidad e interés. Por lo tanto la función de ajuste tiene que incorporar estos criterios.

— **Exactitud de la regla**. Mide la exactitud o precisión de las reglas. Para el caso de reglas de clasificación se utiliza la matriz de confusión. Para el caso de

[4] Lo cual puede suponer la reparación de la regla obtenida, en virtud del algoritmo de reparación expuesto anteriormente.

reglas de predicción se utiliza una matriz similar a la de confusión que se denomina matriz de contingencia. En la bibliografía se ha descrito una gran cantidad de métricas para evaluar la exactitud de la regla a partir de esta matriz de contingencia.

— Comprensibilidad de la regla. La medida más utilizada de comprensibilidad es una medida sintáctica de la longitud de la regla. En general mientras menor es el número de reglas y el número de condiciones en una regla, más comprensible es ésta.

— Interés de la regla. Mide el interés de las reglas. Es la medida más difícil de calcular y existen dos aproximaciones para medir el interés de una regla: aproximación objetiva y aproximación subjetiva. La aproximación objetiva o dirigida por usuario se basa principalmente en tener en cuenta conocimiento o expectativas anteriores del usuario, por lo que es dependiente del dominio. En cambio la aproximación subjetiva o dirigida por datos sólo utiliza los propios datos, por lo que es independiente del dominio. Normalmente se pueden combinar ambos tipos de medidas en lugar de utilizarlos de forma mudamente exclusiva.

Por esta razón, la función de ajuste debe estar formada por un vector de tres valores donde cada uno mida uno de estos criterios. Las métricas que se han seleccionado como objetivos parciales son las denominadas factor de certeza (Shortliffe & Buchanan, 1975), medida de intereses (Tan & Kumar, 2000) y medida de simplicidad (Liu & Kwok, 2000). La elección de dichas medidas se basa en el análisis de la bibliografía disponible y en que cada una de ellas está relacionada con cualidades distintas. En cuanto al factor de certeza, indicar que se trata de una medida de exactitud que, según (Delgado et al., 2001), ha mostrado mejores resultados que la confianza, medida mpleada clásicamente para cuantificar esta cualidad. La medida denominada interesabilidad, que está relacionada con el interés de las reglas, produce, según (Silverstein et al., 1998), mejores resultado que los que arroja el empleo de la medida interés, de la cual deriva. La medida denominada simplicidad (Liu & Kwok, 2000) mide la longitud sintáctica de la regla y está relacionada con la compresibilidad de la regla ya que, según su autor, esta cualidad es inversamente proporcional a su tamaño, y se puede calcular como:

Esta medida puede tomar valores entre 0 y 1, donde el valor máximo 1 indica la regla más simple y el valor mínimo de 0 indica la regla más grande posible.

El uso de métricas combinadas para la resolución del problema de descubrimiento de reglas no suele ser una buena aproximación debido fundamentalmente a que cada una de ellas está asociada a una de las cualidades deseables para las reglas y, con frecuencia dichas cualidades suelen estar en conflicto. Aunque, en general, se ha reportado que el uso de este tipo de métricas produce mejores resultados que de una sola métrica que contemple uno de los objetivos a optimizar, es mucho más correcto trabajar desde el

punto de vista de los algoritmos basados en el concepto de frente de Pareto. En estos algoritmos, existe un vector de objetivos a optimizar por individuo, y el propósito de los algoritmos es hacer que se converja hacia el conjunto que está formado por las mejores soluciones (en términos de todos los objetivos individuales, no de cada uno por separado), denominado frente de Pareto (Freitas, 2002). Se han implementando dos algoritmos pertenecientes a esta familia, el algoritmo MOGA y el NSGA.

5. COMPARACIÓN CON OTROS MÉTODOS CLÁSICOS DE DESCUBRIMIENTO DE REGLAS

Con el objeto de comparar los resultados producidos por algoritmos clásicos con los producidos por los algoritmos basados en GBGP, se han realizado implementaciones para los algoritmos Apriori, PRISM e ID3 orientadas a la generación de reglas de predicción. La elección de estos algoritmos como referente frente a los desarrollados se debe a su popularidad en el ámbito de la minería de datos, que hace hayan sido utilizados en multitud de ocasiones en esta tarea de comparación (Freitas, 2002). Tras analiza las diferencias entre las reglas de asociación, clasificación y predicción; de la discusión allí planteada, puede sacarse como conclusión que las diferencias sintácticas que existen entre dichas tipologías de reglas son mínimas, siendo las principales diferencias entre ellas diferencias de tipo semántico. Por esta razón, no resulta difícil modificar un algoritmo de extracción de reglas de asociación para realizar la tarea de modelado de dependencias, del mismo modo que es posible modificar un algoritmo de extracción de reglas de clasificación para que realice la misma labor.

Todos los algoritmos de GBGP se han implementado en Java utilizando la biblioteca de clases Java para Computación Evolutiva desarrollada por el grupo «Aprendizaje y Redes Neuronales Artificiales» de la Universidad de Córdoba (Ventura et al., 2001). Los algoritmos clásicos empleados se implementaron también lenguaje Java y están integrados en la herramienta EPRules (Romero et al. 2003), una herramienta de extracción de reglas de predicción que ha sido desarrollada para implementar todas las fases del proceso de extracción de conocimiento.

Se han realizado dos tipos de pruebas orientadas a comparar los resultados que produce cada uno de los algoritmos implementados en la tarea de descubrimiento de conocimiento planteada. La primera está orientada a comparar el tiempo de ejecución de los algoritmos. Esta variable es de cierta importancia si se pretende implantar esta fase de extracción de conocimiento en línea con la aplicación. La segunda batería de pruebas está orientada a comparar el número de reglas descubiertas en cada caso y la calidad de las mismas en base a las métricas planteadas anteriormente.

Los resultados obtenidos muestran que, los algoritmos clásicos, y sobre todo el Apriori producen, en general, reglas muy 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,

Con respecto a los algoritmos evolutivos, 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 bastante superior. Además el uso de algoritmos basados en el concepto de frente de Pareto (MOGA y NSGA) permite optimizar, los tres objetivos planteados de forma simultánea, produciendo la mayor proporción de reglas exactas, comprensibles e interesantes.

6. HERRAMIENTA GRÁFICA PARA DESCUBRIMIENTO DE REGLAS DE PREDICCIÓN EN EDUCACIÓN BASADA EN WEB

En la actualidad existen multitud de herramientas tanto comerciales como de libre distribución para la realización de diferentes tareas de minería de datos, entre ellas el descubrimiento de reglas. De entre todas ellas se pueden destacar DBMiner (Klösgen & Zytkow, 2002) y Weka (Witten & Frank, 2000) por ser sistemas de dominio público muy populares, tener un entorno gráfico integrado y permitir realizar casi todas las tareas de minería de datos. El principal inconveniente que presentan este tipo de herramientas es que son complejas de manejar para una persona no experta en minería de datos, además de que al ser de propósito general no se le puede realizar un tratamiento específico del conocimiento de los cursos. Debido a estos problemas, se ha desarrollado una herramienta específica (Romero et al. 2003) que se ha denominado EPRules (Education Prediction Rules) con el objetivo de facilitar el proceso de descubrimiento de reglas de predicción en sistemas educativos basados en web. La principal característica de esta herramienta es su especialización en educación, utilizando atributos concretos, filtros y restricciones específicas para datos de utilización de los cursos.

Utilizando esta herramienta el profesor o autor del curso puede realizar todo el proceso de descubrimiento de conocimiento, desde seleccionar y preprocesar los datos de utilización de los cursos, hasta visualizar las reglas descubiertas al aplicar los algoritmos de minería de datos. Al ser una herramienta gráfica

desarrollada en Java, se necesita tener instalado una máquina virtual de Java para poder ejecutarla. Una vez ejecutada la aplicación aparece su interfaz gráfico (ver Figura 6) que se compone de cuatro partes principales:

— **Datos de Entrada**. Permite abrir una base de datos ya existente con datos de utilización de un curso o bien crear una nueva, y añadirle nuevos alumnos (ver Figura 6). Para crearla o añadir datos se deben seleccionar los ficheros de utilización del curso a preprocesar, que pueden ser tanto la información de un solo alumno (opción Seleccionar Ficheros de la Figura 6) como la de un grupo de alumnos (opción Seleccionar Directorio de la Figura 6).

— Ver Datos. Permite visualizar todos los datos de utilización de los alumnos del curso y realizar algunas estadísticas básicas (máximos, mínimos, medias, etc.) sobre ellos. Estos datos son sobre los tiempos, aciertos o niveles obtenidos por los alumnos en las distintas páginas Web que componen el curso. Se pueden seleccionar desde visualizar los datos o realizar las estadísticas de todos los alumnos, hasta las de un alumno en concreto, o sólo para un tema determinado del curso, o sobre un concepto determinado, o de un nivel de visibilidad o dificultad de un tema determinado, para un número de repetición determinado, y sólo de un tipo determinado (tiempo, acierto o nivel).

— Reglas de predicción. Permite aplicar los distintos algoritmos de descubrimiento de reglas disponibles (ID3, Apriori, Prism y las diferentes versiones de GP), pudiendo seleccionar tanto el algoritmo que se desea

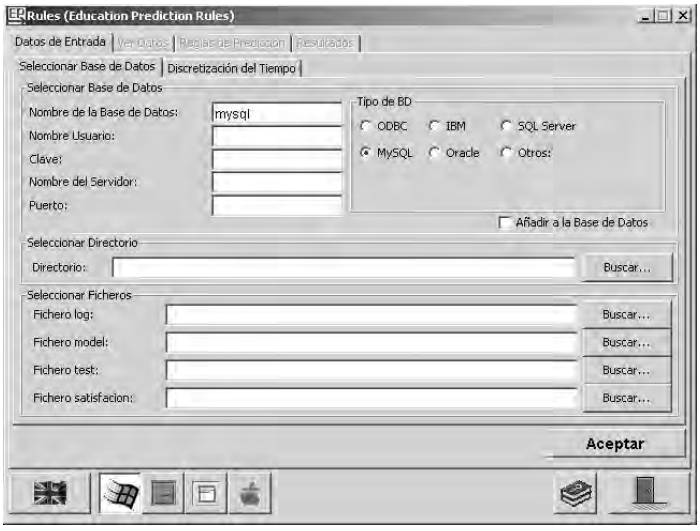


FIGURA 6. Herramienta Gráfica EPRules para el Descubrimiento de Reglas de Predicción.

utilizar y sus parámetros de ejecución específicos (ver Figura 7), como las restricciones subjetivas que deben de cumplir las reglas (ver Figura 8), como las medidas de evaluación objetivas a utilizar como filtrado final de las reglas descubiertas, de manera que las reglas que finalmente se le muestran al usuario le sean realmente de utilidad.

— Resultados. Permite visualizar tanto los datos de utilización de los cursos, como los resultados de las estadísticas, como las reglas de predicción descubiertas (ver Figura 9). En concreto, en el último caso, para cada regla de predicción descubierta se muestran las condiciones que componen el antecedente y el consecuente de la regla, así como todos los valores para cada una de las medidas de evaluación de reglas disponibles. Aunque en principio las reglas se muestran ordenadas por el orden en el que se han descubierto, se pueden ordenar alfabéticamente por una condición del antecedente o por la del consecuente, o numéricamente por el valor de cualquiera de las medidas.

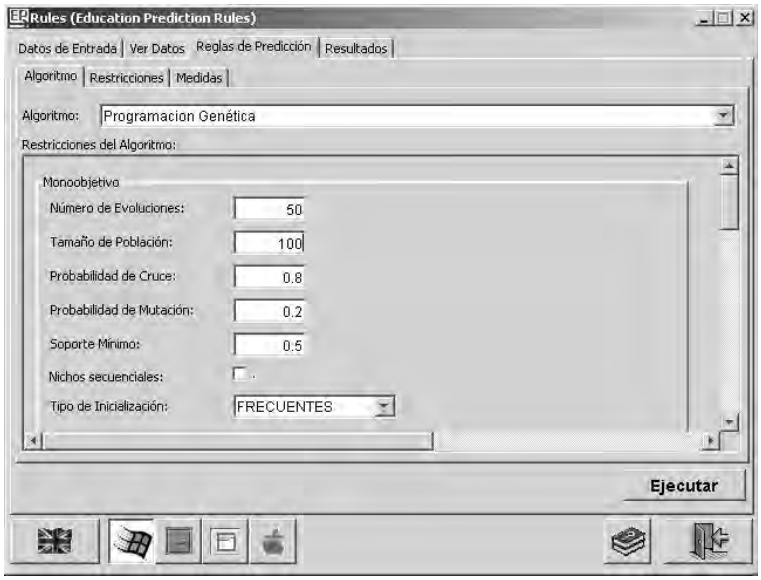


FIGURA 7. Pantalla de selección del algoritmo de descubrimiento de reglas y parámetros.

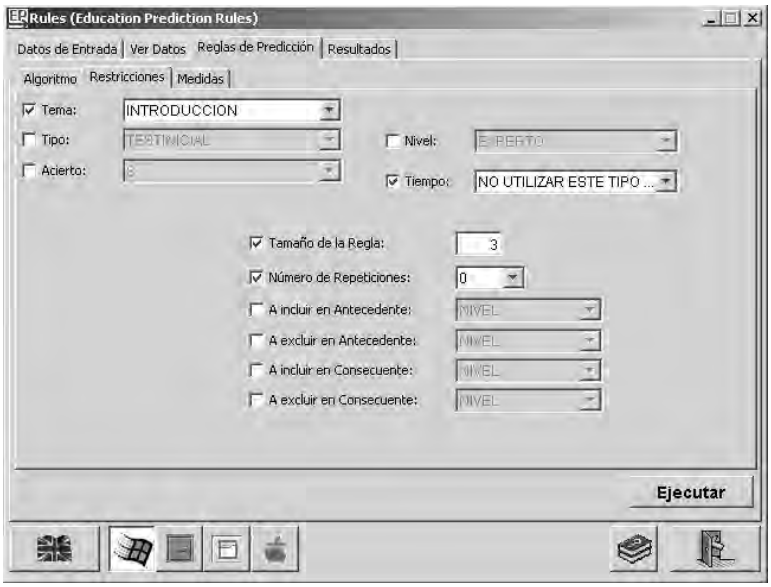
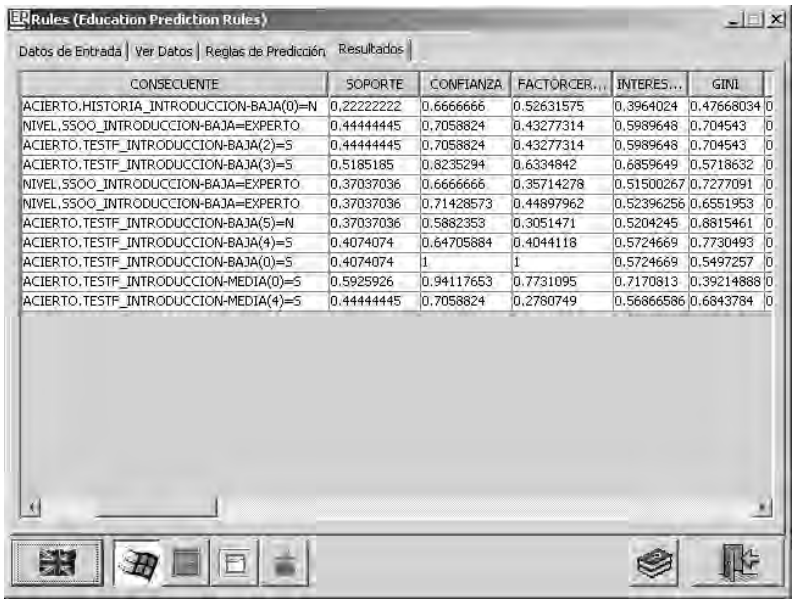


FIGURA 8. Pantalla de selección de las restricciones que deben de cumplir las reglas de predicción.

7. DESCRIPCIÓN DE LA INFORMACIÓN DESCUBIERTA EN FORMA DE REGLAS DE PREDICCIÓN

El objetivo que se persigue con el descubrimiento de las reglas de predicción, es obtener relaciones importantes entre los datos de utilización de los cursos. Esta información descubierta se desea utilizar para la toma de decisiones y deducir qué modificaciones en la estructura de los cursos podrían mejorarlo. Tradicionalmente la información que se utiliza para la evaluación de un curso (Zaïane & Luo, 2001) es la información obtenida al realizar un análisis estadístico de los datos de los alumnos, que sólo tienen en cuenta a los elementos o datos de forma individual, descubriendo información sobre los valores máximos, mínimos, medias, varianzas, etc. El diseñador del curso podría utilizar esta información para tomar decisiones sobre posibles modificaciones a realizar sólo en contenido de las distintas páginas web. Por ejemplo, puede decidir cambiar el contenido teórico de una determinada página debido al excesivo tiempo de visualización de dicha página, o cambiar el enunciado de las preguntas donde el porcentaje de aciertos o fallos sea muy elevado. En cambio la propuesta que se hace en este trabajo consiste en utilizar la información referente a relaciones entre los elementos o datos de utilización, además de la anterior información estadística referente a elementos individualmente. Utilizando esta nueva información se podrían tomar decisiones para realizar modificaciones, no sólo



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 9. Pantalla de Resultados con ejemplo de reglas descubiertas.

sobre el contenido, sino también sobre la estructura del curso. Para ello, el diseñador, a partir de los distintos tipos de reglas de predicción descubiertas, debe obtener sus propias conclusiones sobre las relaciones descubiertas y determinar si se deben realizar modificaciones en el curso para eliminar o reforzar estas relaciones dependiendo de si son beneficiosas o no.

La información descubierta en un proceso de descubrimiento de conocimiento va a depender de la información de partida que se utilice. En el problema concreto de datos de utilización de cursos, estos datos iniciales pueden ser de tres tipos: información relativa a los tiempos de visualización de las páginas, aciertos y fallos obtenidos en las preguntas y niveles de conocimiento en las distintas actividades y test iniciales y finales (tiempo, acierto, nivel). Por lo tanto, la información descubierta va a hacer referencia a estos tres aspectos de los escenarios de un curso adaptativo basados en web:

— Tiempo. El empleado en la realización de las preguntas de las actividades y test iniciales, en la realización completa de los test finales y en la visualización de cada página web de contenido expositivo del curso.

— Acierto. Recogido en las distintas preguntas realizadas en el curso. Tanto los test iniciales y finales como las actividades están formadas por preguntas.

— Nivel. El obtenido tras realizar las distintas actividades y test que componen el curso. En el curso concreto de Linux se han utilizado 3 niveles distintos: principiante, medio y experto.

En principio, la información descubierta va a hacer referencia a distintos tipos de relaciones entre estos tres aspectos . Una regla podría contener elementos o condiciones de cualquiera de estos tres tipos en el antecedente o consecuente de la regla.

A continuación se van a comentar algunos ejemplos de reglas descubiertas a partir de los datos de utilización del curso hipermedia adaptativo basado en web de Linux, y su posible utilización como ayuda para la toma de decisiones sobre la mejora de dicho curso. La mayoría de las reglas de ejemplo que se van a mostrar van a tener un tamaño mínimo, es decir, un solo antecedente y un solo consecuente, para facilitar su compresión.

Un ejemplo de regla descubierta sobre aciertos y fallos en preguntas es:

Si ACIERTO.UNIX_INTRODUCCION-ALTA(2) = SI Entonces
ACIERTO.HISTORIA_INTRODUCCION-BAJA(0) = SI
Factores (Interés = 0.71, Certeza = 0.77, Soporte = 0.59)

Esta regla muestra la relación que existe entre el acierto SI de la pregunta número 2 de la actividad del concepto UNIX que tiene un nivel ALTO en el tema de INTRODUCCION y la pregunta número 0 de la actividad del concepto HISTORIA que tiene un nivel BAJO en el mismo tema. Con respecto a las métricas de evaluación de calidad es una regla muy interesante (interesabilidad de 0.71), exacta (certeza de 0.77) y que afecta a casi el 60% de los alumnos (0.59 de soporte). En este caso particular se consideró que el ítem UNIX_ INTRODUCCION (2) debía tener el mismo grado de dificultad que el ítem HISTORIA_INTRODUCCIÓN (0).

Otro ejemplo de regla descubierta que relaciona el nivel de conocimiento en un concepto y acierto o fallo a una pregunta referente a este concepto es:

Si NIVEL.EMULADORES_PROGRAMAS -ALTA = EXPERTO Entonces
ACIERTO.EMULADORES_PROGRAMAS-ALTA(1) = N
(Interés= 0.69, Factor Certeza= 0.73, Soporte = 0.44)

Esta regla muestra la relación que existe el nivel EXPERTO en el nivel final de evaluación del concepto EMULADORES que tiene dificultad ALTA dentro del tema PROGRAMAS y el acierto NO a la pregunta número 1 de la actividad de evaluación del mismo concepto. Con respecto a las métricas de evaluación esta regla es interesante, exacta y afecta a casi la mitad de los alumnos. 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.

Finalmente, otro ejemplo de regla descubierta que relaciona el tiempo de visualización de una página y el acierto o fallo a una pregunta es:

Si TIEMPO.TESTF_ADMINISTRACION-ALTA(0) = ALTO Entonces
ACIERTO. TESTF_ADMINISTRACION-ALTA(0) = N
(Interés= 0.51, Factor Certeza= 0.79, Soporte = 0.27)

Esta regla muestra la relación entre la visualización con tiempo ALTO de la pregunta 0 del test final de grado ALTO del tema Administración y el acierto NO de dicha pregunta. Con respecto a las métricas de calidad de la regla es exacta, pero menos interesante y afecta a un menor número de alumnos que las reglas anteriores. 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.

8. CONCLUSIONES

En este trabajo se ha presentado una metodología para la mejora de sistemas hipermedia adaptativos educativos basados en web, apoyado en el uso de técnicas de aprendizaje evolutivo, para la extracción de información interesante que puede revertir en dicha mejora.

Se ha propuesto la aplicación de Algoritmos Evolutivos para llevar a cabo la tarea de extracción de conocimiento, mediante descubrimiento de reglas de predicción. En concreto, se ha trabajado en el paradigma de la Programación Genética Basada en Gramáticas, representando cada regla mediante un árbol de derivación de una gramática de contexto libre. Un análisis de las distintas métricas existentes para valorar la calidad de las reglas producidas, revela la necesidad de la aplicación de algoritmos multiobjetivo. En concreto, se han utilizado las aproximaciones MOGA y NSGA. La calidad de los resultados, en función del número de reglas obtenidas, tiempo empleado en la ejecución del algoritmo, y el grado de interés, precisión y comprensibilidad de las reglas, son muy superiores en este caso en comparación con el resto de algoritmos propuestos, que utilizan una única medida o una composición de varias.

Con respecto a la utilidad práctica de las reglas descubiertas para la toma de decisiones sobre posibles modificaciones que se pueden realizar en ASWEs, 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. Finalmente, para facilitar el proceso de descubrimiento de conocimiento se ha desarrollado una herramienta específica 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.

Como línea de trabajo futura se plantea el análisis y desarrollo de nuevas métricas para evaluar el interés de las reglas generadas. Debido a que existen multitud de métricas para evaluar la calidad de las reglas, algunas de las cuales se han empleado en los algoritmos de descubrimiento de conocimiento implementados. Sin embargo, muchos de ellos están relacionados, y resulta difícil

encontrar un conjunto de ellos que reflejen fielmente la calidad de las reglas, sobre todo con aspectos subjetivos de las reglas, como el interés. Un estudio preliminar realizado sobre un conjunto de métricas existentes indica que, un análisis en componentes principales (PCA) de dichas medidas produce 3 componentes que almacenan más de un 85% de la varianza de los datos. Considerando que estas variables no están relacionadas, y que se corresponden con direcciones ortogonales del nuevo espacio generado, sería interesante estudiar el empleo de dichas componentes principales como objetivos a optimizar en el algoritmo multiobjetivo. En esta línea de estudio de nuevas métricas, consideramos también interesante la búsqueda de métricas relacionadas con el interés subjetivo que muestran los profesionales por las reglas generadas. En este sentido, existen referencias de AEs (Williams, 1999) en los que no existe una función de aptitud, sino que los individuos son valorados por un experto en cada ciclo del algoritmo, aunque este enfoque podría ser inaplicable dado el número de reglas que pueden generarse en una iteración del algoritmo. Sin embargo, una primera aproximación interactiva, en la que el tamaño de población sea pequeño, podría arrojar información que se utilizase en el desarrollo de métricas que permitieran de forma efectiva modelar estas preferencias.

9. REFERENCIAS BIBLIOGRÁFICAS

Bayardo, R. J. Agrawal, R. (1999). Mining the most interesting rules. Fifth conference ACM on Knowledge Discovery and Data Mining. SIGKDD.

Deb, K. (2001). Multi-Objetive Optimization Using Evolutionary Algorithms. Wiley.

De Bra, P. Brusilovsky, P. Houben, G. (1999). Adaptative Hypermedia: From System to Framework. ACM Computer Surveys. 31 (4).

Delgado, M. Sánchez, D. Martín-Bautista, M.Vila, M. (2001). Mining Association rules with improved semantics in medical databases. Artificial Intelligence in Medicine. 21.

Freitas, A. (2000) Understanding the crucial differences between classification and discovery of association rules. ACM SIGKDD Explorations. 2(1) 65-69.

Freitas, A. (2002). Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag. 2002.

Herín, D. Sala, M. Pompidor, P. (2002). Evaluating and Revising Courses from Web Resources Educational. ITS 2002, LNCS 2363, 208-218.

Klösgen, W. Zytkow, J.M. (2002). Handbook of Data Mining and Knowledge Discovery. Oxford University Press.

Liu, J.L. Kwok, J.T. (2000).An Extended Genetic Rule Induction. Conf. Evolutionary Computation.

Noda, E. Freitas, A. Lopes, H.S. (1999). Discovering interesting prediction rules with a genetic algorithm. Congress on Evolutionary Computatio. Washington D.C., USA.

ORTIGOSA, A. CARRO, R.M. (2002). Asistiendo el Proceso de Mejora Continua de Cursos Adaptativos. III Congreso Internacional de Interacción Persona-Ordenador. 246-250.

ROMERO, C. VENTURA, S. DE CASTRO, C. HALL, W. HONG, M. (2002). Using Genetic Algorithms for Data Mining in Web-based Educational Hypermedia Systems. Adaptive Hypermedia 2002. Workshop on Adaptive Systems for Web-based Education. 137-142.

ROMERO, C. VENTURA, S. CASTRO, C. DE DE BRA, P. (2003). Discovering Prediction Rules in AHA! Courses. LNCS User Modeling'03.

SHORTLIFFE, E. BUCHANAN, B. (1975). A model of inexact reasoning in medicine. Mathematical Biosciences, 23:351-379.

SILVERSTEIN, A. BRIN, S. MOTWANI, R. (1998). Beyond market baskets:Generalizing association rules to dependence rules. Data Mining and Knowledge Discovery. 2:29-68.

SPILIOPOULOU, M. (2.000). Web Usage Mining for Web Site Evaluation. Communicacion of the ACM.

TAN, P. KUMAR, V. (2000). Interesting Measures for Association Patterns: A Perspective. Technical Report TR00-036. Department of Computer Science. University of Minnnesota.

VENTURA, S. ORTIZ, D. HERVÁZ, C. (2001). Jclec: Una biblioteca de clases java para computación evolutiva. Congreso de Algoritmos Evolutivos y Bioinspirados. 23-30.

WHIGHAM, P.A. (1995). Gramatically-based Genetic Programing. Proceeding of the Workshop on Genetic Programming. 33-41.

G.J., WILLIAMS (1999). Evolutionary Hot Spots Data Mining. An Architecture for Exploring for Interesting Discoveries. Conf on Knowledfe Discovery and Data Mining.

WITTEN, I.H. FRANK, E. (1999). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann.

ZAÏNE, O. R. (2001) Web Usage Mining for a Better Web-Based Learning Environment. Technical Report.

ZAÏANE, O.R. LUO, J. (2001). Towards Evaluating Learners' Behaviour in a Web-Based Distance Learning Environment. Proc. IEEE International Conference on Advanced Learning Technologies.

10. PERFIL ACADÉMICO Y PROFESIONAL DE LOS AUTORES

Cristóbal Romero Morales: Profesor Colaborador en el departamento de Informática y Análisis Matemático de la Universidad de Córdoba.
Líneas de investigación: Sistemas Hipermedia Adaptativos, Sistemas Tutores Inteligentes, Data Mining.
E-mail: ma2romoc@uco.es
Sebastián Ventura Soto: Titular de Universidad en el departamento de Informática y Análisis Matemático de la Universidad de Córdoba.
Líneas de investigación: Algoritmos Evolutivos, Sistemas Hipermedia Adaptativos, Data Mining.
E-mail: sventura@uco.es

Carlos de Castro Lozano: Catedrático de Escuela en el departamento de Informática y Análisis Matemático de la Universidad de Córdoba.
Líneas de investigación: Sistemas Multimedia, Enseñanza a distancia, Tecnologías Adaptativas.
E-mail: cdecastro@uco.es
Departamento de Informática y Análisis Numérico.
Campus Universitario de Rabanales.
Ctra. Madrid-Cádiz, Km. 396,2 14071 Córdoba
Teléfono: +34 57 218630 Fax: +34 57 218630