

Descubrimiento de Reglas de Predicción en Sistemas de e-learning utilizando Programación Genética

Cristóbal Romero, Sebastián Ventura, Cesar Hervás

Universidad de Córdoba, Campus Universitario de Rabanales, 14071, Córdoba, España
{cromero, sventura, cdecastro}@uco.es

Resumen. Este artículo describe la utilización de técnicas de minería de datos en sistemas de e-learning para proporcionar retroalimentación a los desarrolladores de courseware. La información descubierta se muestra en forma de reglas de predicción debido a que son altamente comprensibles y se pueden utilizar directamente en procesos de toma de decisiones. El objetivo es utilizar las reglas para descubrir posibles mejoras en cualquier tipo de curso basado en web y en especial en cursos hipermedia adaptativos basados en web. El método de descubrimiento de reglas que proponemos son los Algoritmos Evolutivos y en concreto la Programación Genética Basada en Gramática (PGBG) con técnicas de optimización multiobjetivo, y se va a comparar con otros algoritmos clásicos de descubrimiento de reglas de asociación, construcción de árboles de decisión y inducción de reglas adaptados para el descubrimiento de reglas de predicción. Para facilitar la realización de todo el proceso de descubrimiento de conocimiento se ha desarrollado una herramienta específica denominada EPRules (Education Prediction Rules) orientada para ser utilizada por el propio autor del curso.

1 Introducción

La aplicación de técnicas de descubrimiento de conocimiento y minería de datos en sistemas educativos basados en web es un área de investigación muy novedosa y prometedora [20]. Esta misma idea ya lleva tiempo utilizándose con gran éxito en los sistemas de comercio electrónico [15]. But while e-commerce objective is to guide clients in purchasing, e-learning objective is to guide students in learning. En la actualidad existen multitud de herramientas tanto comerciales como de libre distribución para la realización de diferentes tareas de minería de datos, principalmente el descubrimiento de reglas. De entre todas ellas se pueden destacar DBMiner [21] y Weka [19] 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 estas 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 puede realizar un tratamiento específico del conocimiento de dominio en particular como el de los Adaptive Systems for Web-based Education (ASWEs) [2]. Debido a estos problemas, se ha desarrollado una herramienta específica que se ha denominado EPRules (Education Prediction Rules)

con el objetivo de facilitar el proceso de descubrimiento de reglas de predicción por los propios educadores [13]. A continuación, se va a describir el proceso de descubrimiento de información implementado en EPRules, después se describe la herramienta y finalmente se presentan las pruebas realizadas, algunos ejemplos de reglas descubiertas y las principales conclusiones.

2 Proceso de Descubrimiento de Conocimiento

El proceso completo de descubrimiento de conocimiento de ASWEs la puede realizar el propio profesor o autor del curso utilizando la herramienta EPRules [13]. Comenzando con la selección y preprocesado de los datos de utilización de los cursos, hasta visualizar las reglas descubiertas al aplicar los algoritmos de minería de datos (ver Figura 1).



Fig. 1. Proceso de Descubrimiento de Conocimiento.

Este proceso de descubrimiento comienza siempre con la selección de la base de datos donde se encuentran almacenados los datos de utilización ya preprocesados del curso que se desea utilizar. Si los datos no se encuentran preprocesados, se debe previamente preprocesar los ficheros *logs* capturados por el curso y almacenarlos en una base de datos. A continuación se debe seleccionar el algoritmo de descubrimiento de conocimiento aplicar y sus parámetros específicos, además de las restricciones tanto objetivas como subjetivas que se desean que cumplan las reglas descubiertas. Tras finalizar la ejecución del algoritmo, se visualiza el conjunto de reglas de predicción descubiertas: los elementos que forman el antecedente y consecuente de la regla, así como las medidas de evaluación de cada regla y se determina si el conjunto de reglas descubiertas son o no útiles, tanto por su número, como por su calidad respecto a las diferentes medidas, como por su significado semántico. Entonces se decide o bien utilizar el conjunto de reglas o un subconjunto de las reglas descubiertas, si se consideran suficientemente interesantes para tomar decisiones sobre posibles modificaciones a realizar en el curso, o bien se vuelve a aplicar el algoritmo con distintos parámetros o restricciones para descubrir un conjunto de reglas más interesante que el actual. Todo este proceso se realiza de una forma directa desde la herramienta gráfica EPRules desarrollada específicamente para el problema de descubrimiento de reglas de predicción en ASWEs.

2.2 Descubrimiento de Reglas de Predicción

Las reglas del tipo Si-Entonces son una de las formas más populares de representación del conocimiento debido a su simplicidad, comprensibilidad y capacidad de expresión [21]. Dependiendo del conocimiento que almacenan, se pueden distinguir diferentes tipos de reglas como son las reglas de asociación, clasificación, predicción, causales, de optimización, etc. De entre ellas, el tipo de regla que vamos a utilizar son las reglas de predicción y el formato concreto de rules in EBNF (Extended Backus Naur Form) que se muestra en la Figura 2.

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

Fig. 2. Formato específico de las Reglas de Predicción.

El objetivo de las reglas de predicción [10] es como su propio nombre indica predecir un atributo objetivo dependiendo del valor de otro grupo de atributos. Esta tarea es sintácticamente similar al descubrimiento de reglas de clasificación donde sólo hay una condición en el consecuente, pero esta condición puede ser cualquier par valor-atributo como en el descubrimiento de reglas de asociación. De forma que se puede considerar como una generalización de las reglas de clasificación [11] o una especialización de las reglas de asociación [1]. Esta tarea de descubrimiento de reglas de predicción se le suele denominar modelado de dependencias y consiste en la predicción de relaciones entre atributos especificados o no por el usuario [7]. Las reglas de predicción son muy populares en minería de datos debido a representan conocimiento a un alto nivel de abstracción y se puede utilizar directamente en procesos de toma de decisiones.

2.2 PGBG para Descubrimiento de Reglas

El descubrimiento de reglas se va a realizar mediante Programación Genética Basada en Gramáticas (GBGP) [17] utilizando técnicas de optimización multiobjetivo [4] que permiten utilizar varios criterios de calidad de las reglas. La Programación Genética basada en gramáticas 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, en nuestro caso todas las posibles reglas de predicción. Se ha elegido este paradigma por

la expresividad que presenta, que va a facilitar enormemente la interacción con el usuario restringiendo dicha gramática para que sólo se generen las reglas solicitadas. El algoritmo evolutivo empleado es el siguiente:

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 \leftarrow generación + 1

Fin Mientras

Devolver el conjunto E

Fin

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 mediante penalización en la función de ajuste) 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.

Finalmente, existen multitud de métricas para valorar la calidad de las reglas [8], [16] cada una de las cuáles se centra en algunos aspectos de las mismas. Sin embargo, no existe una medida que supere claramente a las demás en todos los dominios de aplicación. Debido a esto se ha planteado este problema como un problema de *optimización multiobjetivo* [4]. En este caso, no existiría una única función de aptitud asociada a una medida, sino que existen varias funciones que se desean optimizar simultáneamente. Existen varias formas de abordar el problema de la optimización multiobjetivo con algoritmos evolutivos: un primer enfoque hace uso de *funciones de agregación*, mientras que un segundo enfoque que es el que se ha utilizado hace uso del concepto de *frente de Pareto* donde 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 [6]. métricas de calidad de las reglas a optimizar. La función de ajuste utilizada va estar formada por un vector de tres valores donde cada uno mida uno de los tres aspectos principales del conocimiento descubierto por un algoritmo de minería de datos [7] *exactitud*, *comprensibilidad* e *interés*. Las métricas que se han seleccionado como objetivos parciales son las denominadas *medida de factor de certeza* [14], *medida de interés* [16] y *medida de simplicidad* [9].

2.1 Información de Utilización de los Estudiantes

La información de uso que hemos utilizado para realizar el descubrimiento de reglas de predicción, es la información capturada con los usuarios de un curso sobre el S.O. Linux [13]. Hemos desarrollado el curso de Linux utilizando el sistema AHA! [5] 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. En concreto se le han realizado modificaciones para incrementar su poder de adaptación en educación, mediante la personalización del curso dependiendo del nivel de conocimiento de cada estudiante en particular. En la figura 3 se muestra un mismo tema del curso a tres niveles de dificultad distintos (experto, medio y principiante).

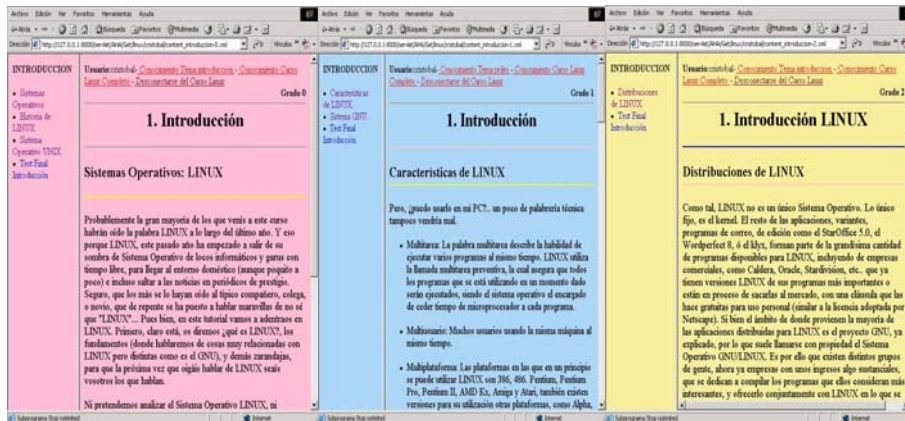


Fig. 3. Ejemplo de tres niveles para el capítulo de Introducción del curso de linux.

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. La información de utilización recogida en los ficheros logs es referente a tiempos de acceso, aciertos en preguntas y niveles de conocimiento de cada alumno. Esta información se ha integrado en una base de datos relacional y se le han realizado una tarea de preprocesado [7]: selección de atributos, limpieza de datos, transformación a valores discretos de atributos continuos e integración de datos.

3 Herramienta EPRules para descubrimiento de reglas

La Herramienta EPRules [12] es una herramienta visual para el descubrimiento de reglas de predicción y está orientada para ser utilizada por el profesor o autor del curso. Se ha implementado en el lenguaje de programación Java y su principal característica es su especialización en educación, utilizando para ello atributos, filtros y restricciones específicas en el dominio concreto de los ASWEs. Es además una

herramienta dinámica, ya que permite añadir fácilmente nuevos algoritmos de descubrimiento de reglas que se implementen y nuevas medidas de evaluación de reglas, con sólo modificar los ficheros de configuración (ficheros java properties) y colocar las nuevas clases en el directorio de los algoritmos o de las medidas de evaluación. El interfaz gráfico de la aplicación EPRules se compone de cuatro ventanas principales:

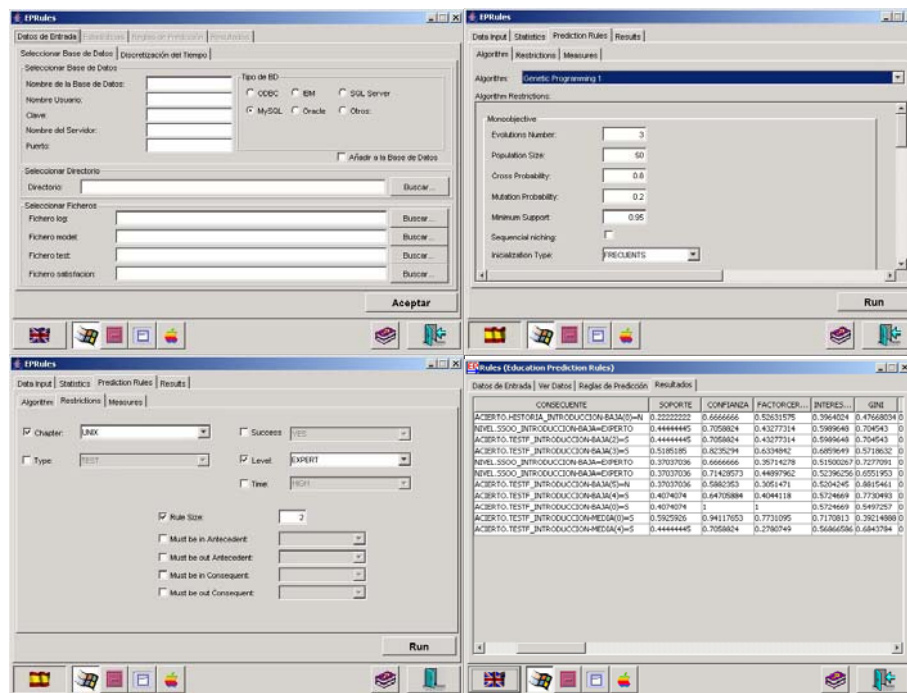


Fig. 4. Diferentes pantallas de la herramienta EPRules.

- **Datos de Entrada.** Desde esta ventana (ver Figura 4 arriba-izquierda) 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. Se han implementando los métodos globales supervisados de igual en frecuencia, igual en anchura y el método manual (donde se especifican los puntos de corte). Sólo se ha discretizado el atributo tiempo, asignándole las etiquetas ALTO, MEDIO Y BAJO.
- **Ver Datos.** Si ya se ha abierto una base de datos de un curso, 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 [11], un algoritmo de reglas de asociación como es el Apriori [1], un algoritmo de inducción de reglas como es el Prism [3] y diferentes versiones de algoritmos evolutivos, en concreto de programación genética basada en gramática [17] con o sin multiobjetivo [4]. Se puede seleccionar tanto el algoritmo que se desea utilizar y sus parámetros de ejecución específicos (ver Figura 4 arriba-derecha), como las restricciones subjetivas que deben de cumplir las reglas (ver Figura 3 abajo-izquierda), de manera que las reglas que finalmente se le muestran al usuario le sean realmente de interés.
- **Resultados.** Esta ventana (ver Figura 4 abajo-derecha) aparece automáticamente después de finalizar la ejecución un algoritmo y permite visualizar todas las reglas de predicción descubiertas (ver Figura 3). 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 [16] [8] 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.

4 Sección Experimental

4.1 Parámetros del Algoritmo Evolutivo

En la Tabla 1 se muestran los parámetros de configuración para los algoritmos que han integrado las pruebas. La inicialización se realiza mediante el método denominado “en rampa”, y el número de producciones se ha ajustado para que se generen individuos con 1, 2 y 3 condiciones en el antecedente.

FASE DE INICIALIZACIÓN	
•	Tamaño de población: 50, 100 y 200 individuos
•	Método de inicialización: En rampa Número mínimo de producciones: 9 Número máximo de producciones: 18
FASE DE REPRODUCCIÓN	
•	Selección basada en orden
•	Operador de cruce selectivo <ul style="list-style-type: none"> - Probabilidad de éxito: 0.8 - Ruleta: <ul style="list-style-type: none"> <antecedente>,<consecuente> = 1 <condición> = 4 <atributo>,<valor> = 2
•	Operador de mutación selectiva: <ul style="list-style-type: none"> - Probabilidad de éxito: 0.2 - Ruleta: <ul style="list-style-type: none"> <antecedente>,<consecuente>,<condición> <atributo>,<valor> = 1
CRITERIO DE PARADA	
•	Número máximo de generaciones: 50

Tabla 1. Parámetros del algoritmo evolutivo.

4.1 Resultados Experimentales

Se han realizado diferentes pruebas orientadas a comparar los resultados que produce cada uno de los algoritmos implementados en la tarea de descubrimiento de conocimiento planteada. El objetivo es comparar el *número de reglas descubiertas en cada caso y la calidad de las mismas* en base a las métricas planteadas anteriormente de exactitud, interés y comprensión. Para la realización de las pruebas con los algoritmos evolutivos, se han ejecutado 10 veces, utilizando los valores medios de todas las ejecuciones. Además se han tres pruebas distintas utilizando todos los datos, sólo los datos frecuentes (aquellos con un soporte superior a 0.5) y por último, los datos dentro de un rango (aquellos con un soporte superior a 0.2 e inferior a 0.9).

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.

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 [1] 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) [4] permite optimizar los tres objetivos planteados de forma simultánea, produciendo en todas las ejecuciones la mayor proporción de reglas exactas, comprensibles e interesantes.

4.1 Descripción 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.

5. Conclusiones y Futuro Trabajo

En este trabajo se ha presentado una herramienta visual para el descubrimiento de conocimiento en forma de reglas de predicción para la mejora de sistemas hipermedia adaptativos educativos basados en web. En concreto, se propuesto la utilización de Grammar Based Genetic Programming con aproximaciones de multiobjetivo. La calidad de los resultados, en función del número de reglas obtenidas y el grado de interés, precisión y comprensibilidad de las reglas, son superiores en este caso en comparación con el resto de 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 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. 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 en ASWEs, de forma que las reglas descubiertas se puedan aplicar directamente sobre el curso, sin la necesidad de 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 AEs [18] en los que no existe una función de aptitud, sino que los individuos son valorados por un experto en cada ciclo del algoritmo.

Agradecimientos

Este trabajo ha sido financiado por el MCYT a través del proyecto TIC2002-04036-C05-02 y de fondos FEDER.

Referencias

1. Agrawal R., Imielinski T., Swami, A.: Mining association rules between sets of items in large databases. ACM SIGMOD International Conference on Management of Data. (1993).
2. Brusilovsky P.: Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies. Int. Conf. on Intelligent Tutoring Systems. (1998).
3. Cendrowska J. PRISM: an algorithm for inducing modular rules. Journal of Man-Machine Studies. 27,349-370. (1987).
4. Coello C.A., Veldhuizen D.A., Lamount G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer. (2002).
5. De Bra P., Ruiters J.: AHA! Adaptive Hipermedia for All. Proc. of the WebNet Conference. 262-268. (2001)
6. Fonseca C.M., Fleming P.J., "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization". Conf. on Genetic Algorithms. San Mateo, CA. 1993.
7. Freitas A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer-Verlag. (2002).
8. Lavrac N., Flach P., Zupan B.: Rule Evaluation Measures: A Unifying View. ILP-99, LNAI 1634. Springer-Verlag Berlin Heidelberg. (1999).
9. Liu J.L., Kwok J.T.: An Extended Genetic Rule Induction. Conf. On Evolutionary Computation. (2000).
10. Noda, E. Freitas, A. Lopes, H.S. Discovering interesting prediction rules with a genetic algorithm. Proc. Congress on Evolutionary Computation, CEC-99. (1999).
11. Quilan J.R.: Generating Production rules from decision trees. Proceeding of IJCAI-87. (1987).
12. Romero C., Ventura S., de Castro C., De Bra P.: Discovering Prediction Rules in AHA! Courses. LNAI 2702 User Modeling'03, 25-34. (2003)
13. Romero C., De Bra P., Ventura S., De Castro C.: Using Knowledge Level with AHA! For Discovering Interesting Relationship. World Congress ELEARN. Montreal. (2002).
14. Shortliffe E., Buchanan B.: A model of inexact reasoning in medicine". Mathematical Biosciences, 23:pp.351-379. (1975).
15. Spiliopoulou M.: Web Usage Mining for Web Site Evaluation. Communication of the ACM. (2000).
16. Tan P., Kumar V.: Interesting Measures for Association Patterns. Technical Report TR00-036. Department of Computer Science. University of Minnesota. (2000).
17. Whigham P.A.: Gramatically-based Genetic Programming. Proceeding of the Workshop on Genetic Programming. pp. 33-41. (1995).
18. Williams G.J.: Evolutionary Hot Spots Data Mining. An Architecture for Exploring for Interesting Discoveries. Conf on Knowledge Discovery and Data Mining. (1999).
19. Witten I.H., Frank E.: Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann. (1999).
20. Zaïene O.R.: Web Usage Mining for a Better Web-Based Learning Environment. Technical Report. (2001).
21. Zytlow J., Klosgen W.: Handbook of Data Mining and Knowledge Discovery. Oxford University Press. (2001).