

# Sistema recomendador colaborativo usando minería de datos distribuida para la mejora continua de cursos e-learning

Enrique García Salcines, Cristóbal Romero Morales, Sebastián Ventura Soto y Carlos de Castro Lozano

**Title**—Collaborative recommender system using distributed rule mining for improving web-based adaptive courses.

**Abstract**—Nowadays, the application of data mining techniques in e-learning and web based adaptive educational systems is increasing exponentially. The discovered useful information can be used directly by the teacher or the author of the course to improve the instructional/learning performance. This can be an arduous task and therefore educational recommender systems are used in order to help the teacher in this task. In this paper we describe a recommender system oriented to suggest the most appropriate modifications to the teacher in order to improve the effectiveness of the course. We propose to use a cyclical methodology to develop and carry out the maintenance of web-based courses in which we have added a specific data mining step. We have developed a distributed rule mining system in order to discover information in the form of IF-THEN recommendation rules about the web courses. We have used an iterative and interactive association rule algorithm without parameters and with a weight-based evaluation measure of the rule interest. And we have used a collaborative recommender system to share and score the obtained recommendation rules in one specific course between teachers of other similar courses and some experts in education. Finally, we have carried out several experiments with real students in order to determine the effectiveness of the proposed system and the utility of the recommended rules.

**Index Terms**—data mining, recommender systems, e-learning, web-based education

Enrique García Salcines es Profesor Colaborador del Dpto. de Informática y Análisis Numérico de la Universidad de Córdoba, CP 14072 España (Teléfono: +34957211020; fax: +34957211051; e-mail: [egsalcines@uco.es](mailto:egsalcines@uco.es)).

Cristóbal Romero Morales es Profesor Contratado Doctor del Dpto. de Informática y Análisis Numérico de la Universidad de Córdoba, CP 14072 España (Teléfono: +34957212172; fax: +34957218630; e-mail: [cromero@uco.es](mailto:cromero@uco.es)).

Sebastián Ventura Soto es Profesor Titular del Dpto. de Informática y Análisis Numérico de la Universidad de Córdoba, CP 14072 España (Teléfono: +34957218349; fax: +34957211051; e-mail: [sventura@uco.es](mailto:sventura@uco.es)).

Carlos de Castro Lozano es Profesor Titular del Dpto. de Informática y Análisis Numérico de la Universidad de Córdoba, CP 14072 España (Teléfono: +34957211020; fax: +34957211051; e-mail: [carlos@uco.es](mailto:carlos@uco.es)).

DOI (Digital Object Identifier) Pendiente

## I. INTRODUCCIÓN

EN los últimos años hemos asistido a un gran incremento de los sistemas de educación on-line o sistemas de e-learning. Cada vez son más los centros de enseñanza públicos o privados que ponen a disposición de sus alumnos sistemas de gestión del aprendizaje (Learning Management Systems, LMS) basados en la web. Los primeros sistemas de este tipo eran de carácter comercial (por ejemplo, WebCT, Virtual-U o TopClass aunque, en la actualidad, cada vez cobran mayor protagonismo sistemas de libre distribución como Moodle, ATutor o ILIAS [1].

Estos sistemas suelen contar con una amplia oferta de actividades y recursos didácticos que se ofrecen indiscriminadamente a todos los alumnos del curso, lo que en multitud de ocasiones conduce a que éstos se vean desbordados ante la oferta disponible y sean incapaces de elegir los recursos que mejor se adaptan a sus características e intereses. Los sistemas adaptativos inteligentes y basados en web para educación (Adaptive and Intelligent Web-Based Educational Systems, AIWBES) aparecen con la intención de solucionar este problema, automatizando la adaptación del sistema a las características y necesidades de los usuarios [2]. Estos sistemas utilizan las técnicas de los sistemas adaptativos [3,4] para construir un modelo de alumno que permite adaptar el sistema a sus necesidades de aprendizaje, y aplican técnicas de inteligencia artificial (IA) [5,6] para automatizar tareas tales como el adiestramiento, la monitorización de los alumnos y el diagnóstico de sus limitaciones. El desarrollo de estos sistemas ha dado lugar a un campo de investigación en auge, que es el aprendizaje colaborativo inteligente, cuyo objetivo es extender la funcionalidad de las herramientas simples de trabajo colaborativo que incorporan los LMS utilizando agentes inteligentes [7].

El desarrollo de un AIWBES es una actividad laboriosa [8], tanto más compleja cuanto mayor es el número de posibilidades de adaptación que se desea ofrecer. Un diseño cuidadoso no suele ser suficiente, es necesario realizar también una evaluación posterior basada en los resultados obtenidos por los usuarios del mismo. Para abordar este problema, se han utilizado con éxito técnicas de

descubrimiento de conocimiento o minería de datos [9] 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 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.

El campo de aplicación de la minería de datos en educación, en particular orientado a los profesores para la mejora de sus cursos, plantea una serie de desafíos a resolver. Por una parte, existe una amplia variedad de cursos e-learning sobre los que se puede aplicar minería de datos, pero los resultados obtenidos con un tipo de curso no necesariamente son válidos o aplicables a otro. La amplia gama de resultados que podría obtenerse dependiendo del tipo de curso, provoca que la búsqueda de patrones generales repetibles que puedan aplicarse a cualquier tipo de curso sea una tarea bastante difícil. Por otra parte, la aplicación de técnicas de minería de datos sobre un curso, de manera concreta y con parámetros específicos de filtrado, podría provocar un problema de descubrimiento de reglas de asociación en bases de datos pequeñas [10], donde la información de partida es insuficiente para construir un modelo que permita inferir comportamientos futuros.

En este artículo proponemos un sistema recomendador colaborativo que permite a profesores y expertos en educación intercambiar experiencias entre sí sobre cómo aprenden sus alumnos, de forma que este conocimiento les permita mejorar sus propios cursos on-line. La sección II presenta los trabajos previos relacionados con nuestra propuesta. En la sección III se describe la arquitectura del sistema y el algoritmo de minería diseñado. Las secciones IV y V describen, respectivamente, la implementación del algoritmo y las pruebas realizadas para evaluar la efectividad del sistema. Por último, se presentan las conclusiones del artículo y las líneas de trabajo futuras.

## II. ANTECEDENTES

A continuación se van a describir los principales antecedentes en las distintas áreas de investigación relacionadas con nuestra propuesta.

### A. Algoritmos de minería de datos

La minería de datos se encuentra enmarcada dentro del proceso de descubrimiento o extracción de conocimiento (Knowledge Discovery in Databases, KDD), entendiéndose como tal a la extracción no trivial de información potencialmente útil, válida, novedosa y comprensible a partir de un gran volumen de datos [11].

Entre los sistemas que aplican técnicas de minería de datos en educación online se pueden destacar: los sistemas de personalización [12] del aprendizaje, los de detección de irregularidades [13] que descubren patrones de navegación irregulares, los que detectan problemas en el diseño y la estructura de cursos de e-learning [14] y los sistemas

recomendadores [15] que clasifican los alumnos y los contenidos para recomendar recursos e itinerarios óptimos.

Una de las técnicas de minería de datos más utilizadas en los sistemas anteriores es el descubrimiento de reglas de asociación. Una regla de asociación [16] del tipo  $X \Rightarrow Y$ , expresa una fuerte correlación entre ítems (atributo-valor) de una base de datos. Se define el *soporte* de una regla como la probabilidad de que un registro satisfaga tanto a su antecedente como a su consecuente. La *confianza* de una regla se define como la probabilidad de que un registro satisfaga al consecuente de la regla habiendo satisfecho el antecedente de la misma.

El problema del descubrimiento de reglas de asociación consiste en encontrar todas las asociaciones que satisfagan ciertos requisitos de soporte y confianza mínimos, los cuales suelen expresarse mediante parámetros que define el usuario. El primer algoritmo que resolvió este problema fue Apriori [16]. En [17] se hace una comparación entre los principales algoritmos para el descubrimiento de reglas de asociación concluyendo que para valores altos del soporte, que aseguran que la confianza obtenida se repita en un futuro, el algoritmo Apriori es el más eficiente. No obstante, el Apriori es muy sensible a los valores de soporte y confianza que elige el usuario, lo cual no es una tarea fácil para alguien no experto en minería.

Una mejora del Apriori es el algoritmo denominado Apriori Predictivo [18], cuya principal ventaja es que el usuario no tiene que especificar los valores umbrales de soporte y confianza mínimos. El algoritmo intenta encontrar las  $N$  mejores reglas de asociación, donde  $N$  es un número fijo, buscando un balance adecuado entre el soporte y la confianza de forma que se maximice la probabilidad de hacer una predicción correcta sobre el conjunto de datos. Utilizando el método bayesiano, se define y calcula un parámetro llamado exactitud predictiva que nos dice el grado de exactitud de la regla encontrada.

### B. Medidas de interés de las reglas descubiertas

Aunque la versión predictiva del algoritmo Apriori representa una ventaja sobre la versión original, el algoritmo no asegura que las reglas obtenidas sean las más interesantes para detectar problemas en el curso e-learning. Por esta razón, es necesario llevar a cabo una evaluación del conocimiento extraído. Tradicionalmente, esta evaluación se ha llevado a cabo mediante medidas objetivas de interés tales como el soporte y la confianza [19], así como medidas procedentes del campo de la estadística tales como Chi-Cuadrado o el coeficiente de correlación, que miden el grado de dependencia entre las variables. Sin embargo, en la actualidad están cobrando una gran importancia las denominadas medidas subjetivas [20], basadas en factores definidos por el usuario.

En [21] se presenta un sistema denominado IAS (*Interestingness Analysis System*) que compara las reglas descubiertas con el conocimiento que tiene el usuario del dominio de interés. A través de un lenguaje de especificación propio el usuario indica la base de



su propia información y las descubiertas por otros usuarios con perfiles similares, que han obtenido dichas relaciones trabajando con sus propias bases de datos. Mediante un procedimiento de valoración subjetiva, los usuarios evalúan el interés de las relaciones obtenidas. De este modo, la base de conocimientos se reforzará con aquellas experiencias que por su peso satisfacen las necesidades de muchos usuarios, lo cual implica recomendaciones cada vez más efectivas.

El sistema de minería de datos distribuida está basado en una arquitectura cliente-servidor con  $N$  clientes que aplican el mismo algoritmo de minería de reglas de asociación de manera local sobre los datos de utilización de un curso online por sus alumnos. Los resultados de este algoritmo, se muestran al profesor en un formato comprensible de tuplas del tipo *regla-problema-recomendación*, para ayudarle a corregir los problemas detectados. Estos resultados puede compartirlos con otros profesores de perfil similar. Veamos en detalle cada elemento de la arquitectura propuesta (Figura 1).

El servidor de aplicaciones contiene dos módulos.

El primero es una aplicación web para gestionar la base de conocimientos (KB) o repositorio de reglas y el segundo módulo es un servicio web que permite que el servidor comparta con el cliente la versión actualizada de la KB en formato PMML [34].

Debido a que inicialmente la KB está vacía, es necesario rellenarla por defecto con un conjunto de tuplas (regla-problema-recomendación) de carácter general y que pueden aplicarse a la mayoría de los distintos tipos de cursos. Para ello, un conjunto de expertos en educación propusieron las primeras tuplas de la KB. Dichos expertos, autorizados por el sistema, pueden añadir, eliminar o editar sus propias tuplas, además de votar por las tuplas propuestas por el resto de expertos.

La aplicación cliente forma parte de una metodología cíclica [35] en la cual un profesor que construye un curso e-learning, pueda ser capaz de detectar posibles problemas en el diseño y los contenidos del mismo, lo cual añade una etapa de retroalimentación o mantenimiento del curso.

En esta metodología se distinguen varias fases: 1) Construcción del curso. El profesor a través de una herramienta autor o utilizando un LMS diseña y desarrolla los contenidos del curso; 2) Ejecución del curso por los alumnos. El sistema, de forma transparente para los usuarios, recoge información que se almacena en una base de datos; 3) Mejora continua. En esta fase, el módulo de minería de reglas trabaja en combinación con la KB para clasificar las reglas encontradas en esperadas, si coinciden con alguna de las presentes en la KB, o inesperadas en caso contrario. Por otra parte, las tuplas inesperadas se ordenan según el algoritmo IAS [21] y, si son valoradas como interesantes por el profesor, pueden ser propuestas para ser analizadas por los expertos y, si así se decidiera, ser insertadas en la KB.

Una vez que el cliente descarga del servidor la versión actualizada de la KB, éste puede aplicar el algoritmo de minería de manera *offline*, o sea sin estar conectado.

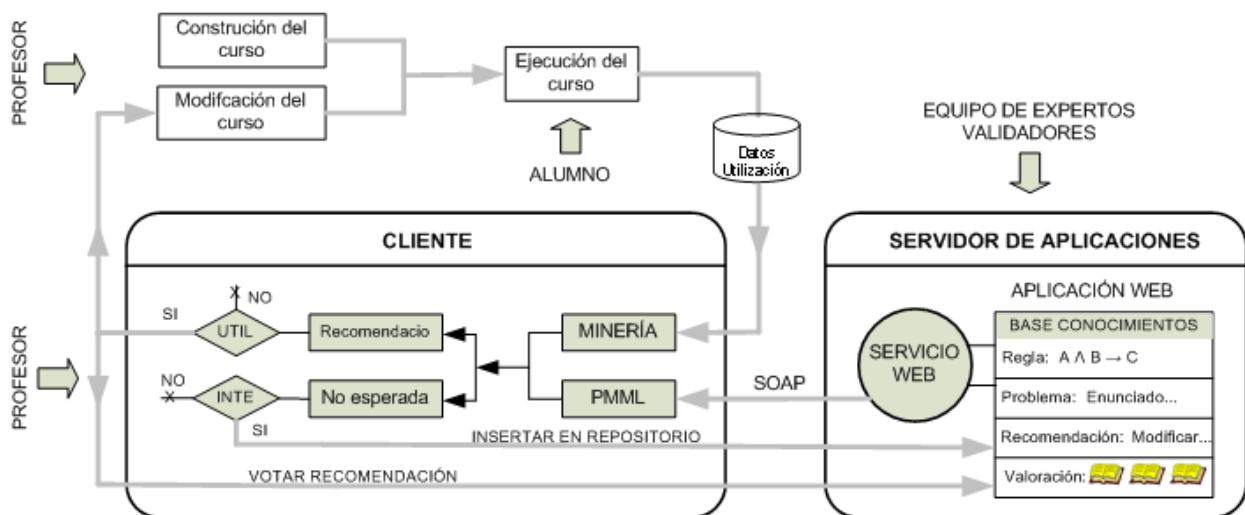


Fig. 1 Arquitectura del sistema CIECoF



número de reglas  $N$  que desea encontrar el usuario; en 2) comienza un bucle cuyas instrucciones se ejecutarán mientras el usuario no decida parar. El paso 3) es el sub-algoritmo al que llamaremos Minería, que describiremos en la siguiente sección, y que devuelve el conjunto de recomendaciones (Rec) y reglas inesperadas ( $R_{ne}$ ) encontradas. De 4) a 6) el usuario vota si le ha sido útil o no la recomendación y de 7) a 11) evalúa las reglas no esperadas para determinar si son interesantes las cuales podrían añadirse previa validación por los expertos en la base de conocimientos KB.

**Entrada:** El perfil de usuario: materia, nivel, estudios, Número de reglas  $N$

- 1) Num =  $N$ ;
- 2) **while** (usuario no desee parar) **do**
- 3)   Rec,  $R_{ne}$  = **Algoritmo\_Minería** (Num);
- 4)   **for each** i-rule in Rec **do**
- 5)     UsuarioVotaRecom(Rec <sub>$i$</sub> );
- 6)   **end**
- 7)   **for each** i-rule in  $R_{ne}$  **do**
- 8)     **if** (Interesante( $R_{ne}$ )) **then**
- 9)       Añadir\_a\_BaseConocimiento( $R_{ne}$ )
- 10)    **end if**
- 11)   **end**
- 12)   Num +=  $N$ ;
- 13) **end while**
- 14) **end all**

Fig. 2. Algoritmo principal

Sean  $U_1, U_2, \dots, U_m$ ,  $m$  usuarios distintos,  $S$  el conjunto de reglas de asociación encontradas por  $U_j$  ( $j=1,2,\dots,m$ );  $S = \{S_1, S_2, \dots, S_m\}$ ; y  $R_1, R_2, \dots, R_n$  todas las reglas en  $S$ ;  $acc(R_i)$  ( $i=1,2,\dots,n$ ) son las exactitudes predictivas de  $R_i$ ;  $R$  conjunto de reglas a descubiertas por el usuario actual,  $R_e$  conjunto de reglas esperadas y  $R_{ne}$  el conjunto de reglas inesperadas tal que  $R = R_e \cup R_{ne}$ ;  $KB$  el conjunto de reglas que conforman la base de conocimientos sobre el dominio. El algoritmo implementado (Fig. 3) está diseñado como a continuación se expone.

En el paso 1) se llama a la función GenRules, que descubre las reglas de asociación, a esta función se le pasa como parámetro el número de reglas deseadas y hace una llamada al algoritmo Apriori Predictivo. El código fuente [37] de este algoritmo ha sido modificado para incluirle restricciones al conjunto de ítems que pueden estar presentes en el antecedente y en el consecuente de las reglas a descubrir.

En el paso 2) se clasifican las reglas encontradas en esperadas, si coinciden sintácticamente con alguna regla de nuestra base de conocimientos, o inesperadas en caso

contrario. En los pasos del 3) al 8) se calcula, para cada regla  $R_i \in R_e$ , la nueva medida de interés basada en pesos  $W_{Acc}$ . En los pasos del 9) al 12) se utiliza el algoritmo IAS para calcular los grados de conformidad de cada regla no esperada  $R_{ne}$  con las reglas almacenadas en la base de conocimientos KB. En 14) se ordena de mayor a menor el conjunto  $R_e$  basado en la medida antes calculada  $AccR$ , mostrando en 15) las recomendaciones correspondientes a cada una de las reglas ordenadas anteriormente. Por último en 16) se le da la posibilidad al usuario de acceder a la unidad de visualización de reglas no esperadas para que este analice cuál de las reglas no esperadas es interesante y posible candidato a incluir en la base de conocimientos.

**Entrada:** el número  $N$  de reglas a descubrir  
**Salida:** Rec: recomendaciones;  $R_{ne}$ : reglas inesperadas

- 1)  $R, acc(R) = \text{GenReglas}(N)$ ; // Predictive APriori
- 2)  $R_e, R_{ne} = \text{ClasificaReglas}(R)$ ;
- 3) **for each** i-rule in  $R_e$  **do**
- 4)  $W_{usuarios_{R_{e_i}}} = \frac{\text{NumVotosUs}(R_{e_i})}{\sum_{j=1}^m \text{NumVotosUs}(R_{e_j})}$ ;
- 5)  $W_{expertos_{R_{e_i}}} = \frac{\text{NumVotosExp}(R_{e_i})}{\sum_{j=1}^k \text{NumVotosExp}(R_{e_j})}$ ;
- 6)  $W_{R_{e_i}} = W_{usuarios_{R_{e_i}}} * C_u + W_{expertos_{R_{e_i}}} * C_e$ ;  
tal que  $C_u + C_e = 1$
- 7)  $W_{acc_{R_{e_i}}} = W_{R_{e_i}} * \frac{\sum_{j=1}^m acc(R_{e_{j,j}})}{m}$ ;
- 8) **end**
- 9) **for each** i-rule in  $R_{ne}$  **do**
- 10)   **for each** j-rule in KB **do**
- 11)     conform <sub>$ij$</sub> , unexpConseq <sub>$ij$</sub> , unexpCond <sub>$ij$</sub> , bsUnexp <sub>$ij$</sub>  = **IAS**( );
- 12)    **end**
- 13) **end**
- 14) **Se ordenan** todas las reglas en  $R_e$  según sus  $AccR$
- 15) **Se muestran** las recomendaciones correspondientes a cada regla  $R_e$
- 16) **Se muestra** un acceso a la unidad de visualización de  $R_{ne}$  según IAS
- 17) **end all**

Fig. 3. Sub-algoritmo Minería

#### IV. IMPLEMENTACIÓN DEL SISTEMA

La aplicación cliente y servidor conforman un sistema híbrido recomendador basado en KBS y CFS, donde las recomendaciones de cambios para mejorar el curso se producen basados en la base de conocimientos que se crea y gestiona en el servidor según los distintos perfiles de usuarios. Se utiliza además, como enfoque complementario el filtrado colaborativo cuyo papel es filtrar y organizar las prioridades de las recomendaciones dependiendo de las votaciones de los expertos y de usuarios de perfil similar.



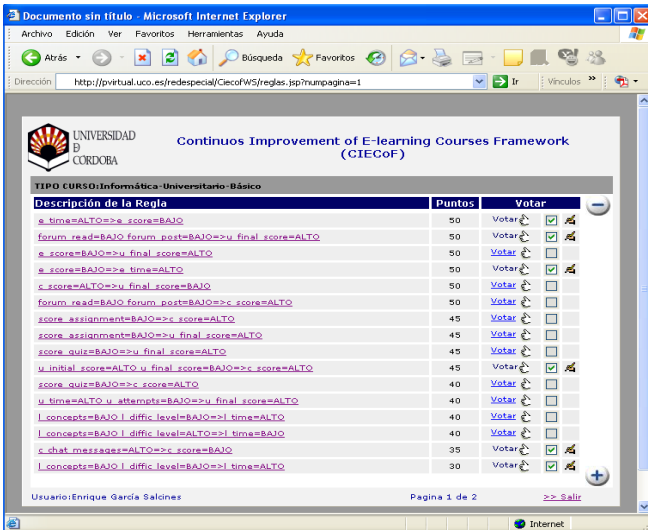


Fig. 5 Interfaz del servidor

V. SECCIÓN EXPERIMENTAL

Para probar nuestra arquitectura necesitábamos datos sobre los que aplicar el proceso de minería. Durante el curso académico 2004-2005 se llevó a cabo en Córdoba la primera experiencia piloto en España para la alfabetización tecnológica de mujeres del entorno rural, denominado “Cordobesas Enredadas”. Para la ejecución del proyecto se desarrollaron 7 cursos correspondientes a los temarios ECDL (Licencia Europea para Manejo del Ordenador), basados en el sistema operativo Guadalinux y el paquete de ofimática OpenOffice.

A. Obtención de datos con INDESAHC

Los cursos se desarrollaron con la herramienta autor INDESAHC [38], que permite la creación de cursos hipermedia adaptativos compatibles con Moodle. La definición del curso está basada en un modelo jerárquico formado por temas divididos en lecciones y donde cada lección contiene una serie de conceptos para la explicación o evaluación de los contenidos de la materia a través de escenarios o páginas web (Figura 6). También se incluye un modelo de adaptación que adapta los contenidos al nivel de conocimiento del alumno. Dicho modelo está basado en un esquema de ocultación de enlaces [4] previa clasificación de los contenidos del curso de acuerdo a distintos niveles de dificultad.

La Tabla I muestra, por una parte, los atributos de datos relacionados con el curso hipermedia adaptativo, que se han añadido como tablas nuevas a la base de datos de Moodle y por otra, otros atributos relacionados con recursos didácticos tales como foros, chats, cuestionarios y tareas que se introducen también desde la interfaz de la herramienta autor.

Una vez que el curso es generado y publicado en Moodle, ya estamos en disposición de utilizar los datos de seguimiento de los alumnos como datos de entrada del proceso de minería.

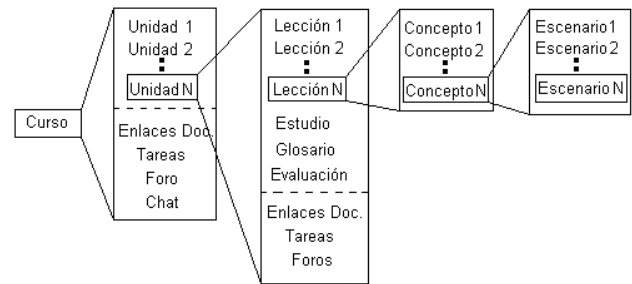


Fig. 6 Modelo del dominio en INDESAHC

TABLA I  
ATRIBUTOS DE DATOS UTILIZADOS EN EL PROCESO DE MINERÍA

Nivel	Atributo	Descripción
Curso	<i>duration</i>	Duración estimada para el curso
	<i>c_time</i>	Tiempo invertido por el alumno
	<i>c_score</i>	Nota media del curso
	<i>c_attempt</i>	Nº intentos antes de aprobar el curso
	<i>c_quiz_attempt</i>	Nº intentos en el cuestionario
	<i>c_quiz_time</i>	Tiempo total en el cuestionario
	<i>c_quiz_score</i>	Nota obtenida en el cuestionario
	<i>c_chat_messages</i>	Nº de mensajes enviados al chat
	<i>assignment_score</i>	Nota de la tarea
	<i>forum_read</i>	Nº de mensajes leídos en el foro
	<i>forum_post</i>	Nº de mensajes puestos en el foro
Tema	<i>u_lessons</i>	Nº de lecciones del tema
	<i>u_time</i>	Tiempo en completar el tema
	<i>u_initial_score</i>	Nota del alumno en el pre-test
	<i>u_final_score</i>	Nota final del alumno en el tema
	<i>u_attempt</i>	Nº intentos antes de aprobar el tema
	<i>forum_read</i>	Nº de mensajes leídos en el foro
Lección	<i>forum_post</i>	Nº de mensajes puestos en el foro
	<i>assignment_score</i>	Nota de la tarea
	<i>doc_view</i>	Si ha consultado el documento
Ejercicio	<i>l_concepts</i>	Nº de conceptos en la lección
	<i>l_time</i>	Tiempo total en completar la lección
	<i>l_diffic_level</i>	Grado de dificultad de la lección
Ejercicio	<i>e_time</i>	Tiempo en completar el ejercicio
	<i>e_score</i>	Nota obtenida en el ejercicio

B. Preprocesado de datos

Antes de aplicar minería de reglas de asociación es necesario preprocesar antes los datos de entrada para adaptarlos a nuestro modelo de datos. Este preprocesado incluye una serie de etapas como la limpieza de datos, la transformación de variables continuas a discretas y la integración de datos cuando estos provienen de distintas fuentes. En nuestro sistema, la limpieza de datos se realiza debido a dos causas muy comunes. Por una parte, se descubrió que el atributo tiempo en muchos casos contenía valores demasiados altos debido a que el alumno ha





descubrió que existía un problema en ese ejercicio del curso hipermedia adaptativo, que pertenecía al tema “Uso de la aplicación”, la lección “Primeros pasos con el procesador de texto” y concepto “Renombrar y guardar un documento”, que era un escenario de INDESAHC de tipo video interactivo donde el alumno debe simular utilizando el ratón los pasos necesarios para completar una actividad. En este caso particular, se comprobó que el enunciado de la pregunta era ambiguo y podía interpretarse de varias maneras, con lo cual se corrigió. Otras reglas de formato similar se encontraron pero relacionadas con preguntas de tipo test o de relación de columnas.

2) Si (u\_forum\_read [2] = BAJO) Y (u\_forum\_post [2] = BAJO) entonces (u\_final\_score [1] = ALTO), exact = 0.75

Esta regla significa que, si los mensajes enviados y leídos del foro 2 que pertenece al tema 1 son bajos, entonces la nota del tema es alta. La regla descubre que ese foro del tema 1 no es necesario o que hay problemas con el tutor. Este tipo de regla descubierta cuestiona la necesidad de un foro a determinados niveles de la jerarquía del dominio, de hecho en nuestro caso se optó por eliminar el foro.

#### E. Análisis de la efectividad de las recomendaciones

Para verificar que los cambios realizados por el profesor tras escuchar las recomendaciones propuestas por el sistema son efectivas, debemos tener en cuenta dos puntos de vista: 1) el del profesor, respecto a qué porcentaje de los problemas supuestamente corregidos, basados en las recomendaciones iniciales, se repiten en sucesivas ejecuciones del curso con grupos de alumnos distintos y 2) el de los alumnos, respecto a cómo influyen las recomendaciones de cambios que ya no se repiten, en las calificaciones finales que éstos obtienen. De estos dos aspectos mencionados se derivan dos hipótesis de partida, en primer lugar tenemos que si los cambios realizados por el profesor son 100% efectivos, estos problemas no debían detectarse de nuevo al analizar los datos de uso en los grupos siguientes; en segundo lugar, si estos no se repiten debían implicar además, una mejora en las calificaciones.

En una etapa de mejora continua del curso como la que proponemos, con sucesivas correcciones en base a los datos de utilización del curso por distintos grupos, sea  $TotalRec_1$  el total de recomendaciones mostradas cuando se analizaron los datos de uso del grupo 1 y que indujeron a cambios en la estructura o contenidos del curso; sea  $TotalRec_{1,i}$  el total de recomendaciones que se repiten en consecutivas ejecuciones del curso con otros grupos de alumnos respecto a las obtenidas con el grupo inicial, podemos calcular la efectividad de los cambios realizados, basándonos en las recomendaciones propuestas en la fase inicial 1 (primera ejecución del curso) respecto a la fase  $i$  ( $i=2,3,...N$ ) correspondiente a consecutivas ejecuciones del curso como:

$$EfectRec_i = \frac{TotalRec_1 - TotalRec_{1,i}}{TotalRec_1} \quad (2)$$

Podemos medir la efectividad que ha tenido en los alumnos la corrección de los problemas detectados, comparando la nota media y desviación estándar en las sucesivas ejecuciones del curso.

Para calcular (2) y comparar las notas finales de los alumnos, se eligió como base material de estudio el curso “Hoja de Cálculo”, y dos grupos de 45 alumnos que harían el curso de manera consecutiva. Con el objetivo de eliminar la influencia de factores externos como conocimientos previos de informática, edad media del grupo, nivel de estudios, etc, que pudiesen alterar el resultado de la investigación, se forzó la composición de los grupos de forma que se cumpliesen los siguientes requisitos: 1) que los alumnos no tuviesen conocimientos previos de informática, esto era relativamente fácil, pues los cursos como hemos mencionado van destinados a la alfabetización digital en entornos rurales; 2) la edad media del grupo fuese la misma; 3) el nivel de estudios no superase el grado medio.

La Tabla II muestra los resultados desde el punto de vista del profesor al aplicar nuestro sistema de manera consecutiva sobre los datos de utilización de los tres grupos de alumnos. La columna “Nuevas” se refiere a las recomendaciones iniciales que da el sistema a problemas detectados en el curso y que el profesor ha considerado útiles y aplicables; la columna “Rep” se refiere a aquellas recomendaciones iniciales que a pesar del profesor haberlas aplicado, se vuelven a repetir las mismas tuplas en ejecuciones consecutivas del curso. La Tabla III muestra los resultados desde el punto de vista del alumno. La columna “NRep” se refiere las tuplas que no se repiten, se muestran además las notas medias finales y desviaciones estándar de cada grupo y se calculan los valores de  $p$ -value comparando el grupo 1 con el grupo 2.

TABLA II  
RESULTADOS DESDE EL PUNTO DE VISTA DEL PROFESOR

Grupo	Nuevas	Repetidas	Total	EfectRecom (%)
1	21	0	21	0
2	5	6	11	72,7

TABLA III  
RESULTADOS DESDE EL PUNTO DE VISTA DEL ALUMNO

Grupo	No Rep.	Nota	p-value 1-2
1	0	6,55 +0,30	
2	15	6,95 +0,56	< 0,0001

Del análisis de los datos de las Tablas II y III se pueden extraer varias conclusiones:

1) Tal y como se suponía en nuestra hipótesis inicial, el porcentaje de efectividad se acerca al 100 % en la medida que el curso se ejecuta más veces. Se detectó que los problemas que se repiten se debieron a cambios en el diseño



- [27] Eliassi-Rad, T. and Shavlik, J., "A System for Building Intelligent Agents that Learn to Retrieve and Extract Information," *International Journal of User Modeling and User-Adapted Interaction*, special issue User Modeling and Intelligent Agents. 13 (4), No. 1-2, 2003, pp. 35-88.
- [28] Geyer-Schulz, A. et al., "An Architecture for Behavior-Based Library Recomendador Systems," *Information Technology and Libraries*. 22(4), 2003, pp.165-174.
- [29] Rosta F., Brusilovsky, P., "Social navigation support in a course recommendation system", *Adaptive Hypermedia and Adaptive Web-Based Systems: 4th International Conference, AH 2006*, 2006, pp. 91-100.
- [30] Tang T., McCalla, G., "Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment". *International Journal on E-Learning*, 4(1), 2005, pp. 105-129.
- [31] Terveen, L. and Hill, W., "Beyond Recomendador Systems: Helping People Help Each Other". In *J. M. Carroll (Ed.) Human-Computer Interaction in the New Millennium*, Addison-Wesley. ACM Press, New York, ch 22, 2001, pp. 487-509.
- [32] Burke, R., "Semantic ratings and heuristic similarity for collaborative filtering", In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, Texas, July 30th–August 3<sup>rd</sup>, 2000.
- [33] Burke, R., "Knowledge-based Recomendador Systems". In *A. Kent (ed.), Encyclopedia of Library and Information Systems*. Vol. 69, Supplement 32. New York: Marcel Dekker. 2000.
- [34] Data Mining Group. Predictive Model Markup Language (PMML). Disponible en Abril de 2008 en <http://www.dmg.org/pmml-v3-0.html>.
- [35] García, E., Romero, C. et al, "Using Rules Discovery for the Continuous Improvement of e-Learning Courses", en *Proc. of the 7th International Conference on Intelligent Data Engineering and Automated Learning- IDEAL 2006*, LNCS 4224, 2006, pp. 887-895
- [36] Good I., *Probability and the weighting of evidence*. Charles Griffin, London, 1950.
- [37] Paquete de minería de datos Weka. Disponible en Abril de 2008 en <http://www.cs.waikato.ac.nz/ml/weka/>.
- [38] De Castro, C., García, E., Romero, C., Ventura, S. "Herramienta autor INDESAHC para la creación de cursos hipermedia adaptativos". *Revista latinoamericana de tecnología educativa*. Vol. 3, 1, 2004.

- [39] H. Liu, F. Hussain, C.L. Tan, and M. Dash. "Discretization: An enabling technique". *Journal of Data Mining and Knowledge Discovery*, 2002, pp. 393-423.



**Enrique García Salcines** es Profesor colaborador del Departamento de Informática de la Universidad de Córdoba en España. Actualmente, está realizando su Tesis Doctoral en el campo de minería de datos aplicada a educación. Su área de interés principal es la aplicación de técnicas de inteligencia artificial para la mejora del aprendizaje.



**Cristóbal Romero Morales** es Profesor Contratado Doctor del Departamento de Informática de la Universidad de Córdoba en España. Es doctor en Informática por la Universidad de Granada desde el año 2003. Su área de interés principal es la aplicación de minería de datos en educación.



**Sebastián Ventura Soto** es Profesor Titular del Departamento de Informática de la Universidad de Córdoba en España. Es doctor en Ciencias por la Universidad de Córdoba desde el año 1996. Su área de interés principal es soft-computing y sus aplicaciones.



**Carlos de Castro Lozano** es profesor titular del departamento de Informática de la Universidad de Córdoba en España. Es doctor en Ciencias por la Universidad de Córdoba desde el año 1983. Sus principales áreas de interés son metodologías y recursos en e-learning y accesibilidad.