Collaborative recommender system using distributed rule mining for improving web-based adaptive courses

E.García, C.Romero, S.Ventura, C.de Castro
{egsalcines, cromero, sventura, cdecastro}@uco.es
University of Córdoba,
Campus de Rabanales, Ctra Madrid-Cádiz Km 396,2
14071 Córdoba, Spain
Tel. (34) 957211020
Fax (34) 957211051

D. Enrique García Salcines is an Assistant Professor in the Computer Science Department of the University of Córdoba, Spain. Nowadays, he is doing the Ph. D. Thesis in the field of educational data mining and his research interests lie in e-learning improvement and data mining techniques.

Dr. Cristóbal Romero is an Assistant Professor in the Computer Science Department of the University of Córdoba, Spain. He received his Ph. D. in Computer Science from the University of Granada in 2003. His research interests lie in artificial intelligence in education and data mining.

Dr. Sebastián Ventura is an Associate Professor in the Computer Science Department of the University of Córdoba, Spain. He received his Ph. D. in the Sciences from the University of Córdoba in 1996. His research interests lie in soft-computing and its applications.

Dr. Carlos de Castro Lozano is an Associate Professor in the Computer Science Department of the University of Córdoba, Spain. He received his Ph. D. in the Sciences from the University of Córdoba in 1983. His research interests lie in e-learning methodologies and resources, as well as soft-computing and accessibility.

Abstract. Nowadays, the application of data mining techniques in e-learning and web based adaptive educational system 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 are going to describe a recommender system oriented to suggesting 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 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.

Keywords: data mining, association rule mining, recommender systems, e-learning, web-based adaptive education

1. Introduction

Recently, the huge increase in Internet access has made the concept of online education or e-learning a reality (Itmazi, J.A.S., 2005). This is a form of computer-aided instruction virtually independent of a specific location and any specific hardware platform (Brusilovsky, P., 2003). Public and private schools are increasingly providing their students with e-learning systems that are also called Learning Management System (LMS), Course Management System (CMS), Learning Content Management System (LCMS), Managed Learning Environment (MLE), Learning Support System (LSS) or Learning Platform (LP). WebCT, Virtual-U, TopClass are a few example of commercial LMS, although open source systems such as Moodle, ATutor and ILIAS are gradually becoming more widespread. Comparative studies between LMS can be found in (Itmazi, J.A.S., 2005). Although LMS provides useful tools for computer-supported collaborative learning, such as forums, chat rooms, discussion groups and e-mail, most of them display contents and educational material in the same way to all students, allowing them to choose their own learning pathway through the course, which is not necessarily the most effective one in terms of their previous knowledge or needs.

One possible solution to this problem is to use Adaptive and Intelligent Web-Based Educational Systems (AIWBES) (Brusilovsky, P., 2003), which combine the techniques of adaptive systems (Brusilovsky, P., 1996; De Bra, P. & Calvi, L.:1998). These systems build a model of the objectives, preferences and knowledge of an individual user in order to adapt the system to his or her learning needs, by means of Artificial Intelligence (AI) techniques from intelligent systems (Brusilovsky, P., Schwarz, E., & Weber, G.,1996; Heift, T., & Nicholson, D.,2001) such as machine learning, Data Mining (DM) and intelligent agents. Hence, certain activities can be performed by the system, which were traditionally carried out by the teacher, such as training and monitoring the students, and diagnosing their limitations.

Many of the systems mentioned above use data mining techniques in order to personalise the output data obtained, avoiding information overload and recommending items required by the current user based on previous or current interactions of other users with similar profiles (Costaguta R., 2006). Recommendation systems assists the natural process of relying on friends, classmates, lecturers, and other sources of make the choices for learning (Lu, J., 2004). In the educational setting, these recommendations systems can be classified according to their field of application or focus (Romero, C. and Ventura, S., 2006): 1) student-centred (Gaudioso E., Santos O., Rodriguez A., y Boticario J., 2003; Zaiane O., 2002), in order to suggest good learning experiences for the students in accordance with their preferences, needs and level of knowledge; and 2) teacher-centred (Chen W. & Wasson B.:2002; Romero, C., Ventura, S., Bra, P. D., & de Castro, C.,2003), with the aim of helping the teachers and/or authors of the e-learning systems to improve the functions or performance of these systems based on user information. Some others examples of educational applications of these systems are: obtaining more feedback about teaching; finding out more about how students learn on the web; evaluating students in terms of their browsing patterns; classifying students into groups; or restructuring the contents of the website in order to personalise the course.

The application of data mining to the field of Education, particularly the teacher-centred approach aimed at improving courses, implies a series of hurdles that need to be overcome (Romero, C. and Ventura, S., 2006). On the one hand, there is a wide variety of e-learning and web-based adaptive courses to which data mining can be applied, influenced by three key aspects: firstly, the area of knowledge to which the course relates; secondly, the level of education, in other words whether the course is aimed at university students, secondary or primary school level, special education or any other kind of training; and finally the level of difficulty of the course, the example if it is a basic or beginner course, intermediate, advanced or expert. The wide range of results that can be obtained, depending on these factors, means that it can be fairly tricky to search for general repeatable patterns which could be applied to any type of course. Furthermore, applying data mining locally with specific filtering parameters can

cause problems with association rule discovery in small databases (Zhang C., Zhang S., 2002) where the starting information is insufficient to construct a model that will infer future behaviour.

This paper proposes a collaborative recommender system which would allow teachers and education experts to swap patterns and experiences about how their students learn; this shared knowledge about a specific type of course will then enable them to improve their own e-learning courses. In the context of this paper, personalisation implies using data mining techniques to show teachers recommendations aimed at improving their courses, which fit with their user profile. The paper is structured as follows. The next section reviews previous related studies. Section 3 describes the system architecture and the design of the mining algorithm. Section 4 discusses the implementation of the algorithm, and section 5 describes the experimental trials carried out in order to evaluate the effectiveness of the system. Finally, the last section presents the conclusions of the paper and future areas for research.

2. Related Works

Data mining is part of the Knowledge Discovery in Databases (KDD) process, and is understood to be the non-trivial extraction of previously unknown and potentially useful, valid and comprehensible information from a large volume of data (W. Klösgen, J.M. Zytkow., 2002). There are different types of systems that had applied successfully data mining techniques to online education such as learning personalisation systems (Srivastava, J.; Mobasher, B.; Cooley, R., 2000) for adapting the course to each student, outliers detection systems (Barnett, V.; Lewis, T., 1994) for discovering irregular browsing patterns, evaluation systems (Romero, C., Ventura, S., Bra, P. D., & de Castro, C., 2003) for detecting problems in the design and structure of e-learning courses, and recommender systems (Li, J.; Zaïane, O.R., 2004), for classifying students and contents in order to recommend optimum resources and pathways.

One of the most commonly used data mining techniques in the above-mentioned systems is association rules discovery (Agrawal, R., et al., 1996). The use of association rules is one of the most popular ways of representing discovery knowledge. These rules describe a close correlation between frequent items in a database. An $X \Rightarrow Y$ type association rule expresses a close correlation between items (attribute-value) in a database. The support S of a rule is defined as the possibility that an entry satisfies both X and Y. Confidence is defined as the probability that an entry satisfies Y given that it satisfies X. Therefore the aim is to find all the association rules that satisfy certain minimum support and confidence restrictions, with parameters specified by the user. The first and most popular algorithm is the Apriori (Agrawal, R., et al., 1996), although there are currently many different association rule discovery algorithms available (Zheng Z., et al., 2001). An important improvement to the Apriori algorithm for using in educational environments is the Predictive Apriori (Tobias S., 2001), due to it does not require that the user has to specify any parameter (neither the minimum support threshold nor confidence values). The algorithm aims to find the N best association rules, where N is a fixed number. It strikes an appropriate balance between support and confidence so that it maximises the probability of making an accurate prediction about the dataset. In order to achieve this, a 'predictive accuracy' parameter is defined and calculated, using the Bayesian method, which provides the accuracy of the rule found.

Association rule mining algorithms normally discover a huge quantity of rules and do not guarantee that all the rules found are relevant. Therefore, they must be evaluated in order to find the most interesting rules for a specific problem. Traditionally, the use of objective interestingness measures has been suggested (Tan P., Kumar V., 2000), such as support and confidence, mentioned previously, as well as purely statistical measures such as the chi-square statistic and the correlation coefficient in order to measure the dependency inference between data variables. However, subjective measures are becoming increasingly important (Silberschatz, A., Tuzhilin, A., 1996), in other words measures that are based on

subjective factors controlled by the user.

Most of the subjective approaches involve user participation in order to express, in accordance with his or her previous knowledge, which rules are of interest. (Liu B., Wynne H., Shu C. Yiming M., 2000) proposed an Interestingness Analysis System (IAS), which compares rules discovered with the user's knowledge about the area of interest. Using their own specification language, they indicate their knowledge about the matter in question, through relationships between the fields or items in the database.

In general, frequent item sets are useful for discovering association rules in large databases. However, when working with separate relatively small databases, it is essential to learn how to use experience, common sense and the models created by other people who have already worked with these databases in the past (W. Klösgen, J.M. Zytkow., 2002). Distributed Data Mining (DDM) assumes that data are distributed in two or more sites and that these sites cooperate in order to obtain global results without revealing the data from each site or part of these data. DDM algorithms have been proposed for partitioning the date into subsets (Savasere A., Omiecinski E., and Navathe S. B., 1995; Scheuermann P., 2001). Parallel data mining algorithms have been also proposed to work with large data sets, by dividing them and distributing them between the different processes of a virtual machine. One of the most intuitive methods for finding association rules in a distributed way was suggested by (Cheung D. W., Ng V. T., Fu A.W., Fu Y., 1996); it is known as horizontal data partition, where the mining process is applied locally and the results obtained from each side combined in order to find rules that comply with most of the local databases. In addition to these DDM tools, there are proactive methods that use tools to support collaborative work: this multidisciplinary development normally involves experts from different areas of knowledge such as: knowledge engineers, in charge of modelling knowledge; knowledge database developers, who construct, organise, annotate and maintain these databases; and teams of validating experts, who validate elements of knowledge before they are entered into the contents repository. Collaborative Recommender Systems are based on opinions provided by experts and users, through explicit or implicit voting systems. The main goal is to suggest better solutions based on overall experience.

Recommender Systems (RS) are currently applied to many web based sectors, for example, in e-commerce in order to offer client-personalised services (Zan, H. et al., 2004); in webpage search engines in order to avoid information overload (Eliassi-Rad, T. and Shavlik, J., 2003); and in digital libraries in order to help users find books articles in accordance with their preferences (Geyer-Schulz, A. et al., 2003). Another recent field of application for RS, which is currently booming, is e-learning (Rosta F., Brusilovsky, P., 2006; Tang T., McCalla, G., 2005). E-learning uses different recommendation techniques in order to suggest online learning activities or optimum browsing pathways to students, based on their preferences, knowledge and the browsing history of other students with similar characteristics. Recommendation techniques can be classified in a number of ways (Terveen, L. and Hill, W., 2001). Classifications are based on data sources - on the back of which recommendations are made - as well as the use made of these data. The Collaborative Filtering System (CFS), also referred to as social filtering, depends on a product database, as well as demographic data and potential consumer evaluations of certain products that have not yet been tried. This is perhaps the most familiar, widespread and fully developed of all the recommendation techniques (Burke, R., 2000a). The main idea of CFS revolves around computerising the “word of mouth” process, by which people recommend products or services to each other. If users need to choose between various options when they have no experience of them, they are likely to trust the opinions of those who do have experience. *Knowledge-Based Recommendation* (KBR), on the other hand, aims to suggest objects based on inferences about the user's preferences and needs. Unlike other techniques, it has prior functional knowledge about how a particular item can satisfy a user's needs and therefore can make reasoned judgements about the relationship between this need and a possible recommendation. The user profile can be any knowledge structure that supports this inference. In the case of Google, this would simply be the query entered by the user. In other cases, it might be a

more detailed representation of the user's needs. The Entree system (Burke, R., 2000b) uses *Case-Based Reasoning* (CBR) techniques to make recommendations based on knowledge.

There are several specific works about the application of these techniques (association rule mining and recommender systems) in e-learning systems. (Wang, F., 2002) develops a portfolio analysis tool based on associative material clusters and sequences among them. This knowledge allows educators to study the dynamic browsing structure and to identify interesting or unexpected learning patterns. To do this, he discovers two types of relations: association relations and sequence relations between documents. (Minaei-Bidgoli, B., Tan, P., & Punch, W., 2004) propose mining interesting contrast rules for web-based education systems. Contrast rules help one to identify attributes characterizing patterns of performance disparity between various groups of students. (Markellou, P., Mousourouli, I., Spiros, S., & Tsakalidis, A., 2005) propose an ontology-based framework and discover association rules, using the Apriori algorithm. The role of the ontology is to determine which learning materials are more suitable to be recommended to the user. (Zaïane, O., & Luo, J., 2001) propose the discovery of useful patterns based on restrictions, to help educators evaluate students' activities in web courses. (Li, J., & Zaïane, O.R., 2004) also use recommender agents for e-learning systems which use association rule mining to discover associations between user actions and URLs. The agent recommends online learning activities or shortcuts in acourse web site based on a learner's access history. (Lu, J., 2004) uses association fuzzy rules in a personalized e-learning material recommender system. He uses fuzzy matching rules to discover associations between student's requirements and a list of learning materials. (Romero, C., Ventura, S., & Bra, P. D., 2004) propose to use grammar-based genetic programming with multi-objective optimization techniques for providing a feedback to courseware authors. They discover interesting association rules from student's usage information. (Merceron, A. & Yacef, K., 2004) use association rule and symbolic data analysis, as well as traditional SQL queries to mining student data captured from a web-based tutoring tool. Their goal is to find mistakes that often occur together. (Freyberger, J., Heffernan, N., & Ruiz, C., 2004) use association rules to guide a search for best fitting transfer model of student learning in intelligent tutoring systems. The association rules determine what operation to perform on the transfer model that predicts a student's success.

3. Proposed System and Methodology

In order to tackle the problems discussed in the introduction section, we are going to propose a collaborative recommender system applied to education and aimed at helping the teachers to continually improve their adaptive e-learning courses. Distributed data mining techniques will be used in a hybrid recommender system based on CFS and KBR in order to add a feedback stage in two ways. Firstly, collaborative filtering will benefit from the new interesting relationships discovered by teachers with similar profiles working with their own databases; these interesting or useful relationships will be made available to other teachers so that they can assess them in terms of their relevance or interest for them. Secondly, the knowledge database will be strengthened with experiences that, because of their importance, satisfy the needs of many teachers and therefore give rise to increasingly effective recommendations.

The distributed data mining system is based on a client-server architecture with *N* clients, which applies an association rule mining algorithm locally on the usage data of an online course by students. The results of this algorithm are shown to the teachers in the form of rule-problem-recommendation type tuples in order to help them correct the problems detected. These results can be shared with other teachers with a similar profile in order to create increasingly effective recommendations. Below is a detailed analysis of each element in the proposed architecture (Figure 1).

Two modules are included in the application server. The first is a web application server so the experts can manage a knowledge base (KB), and can add, delete or edit tuples, as well as being able to vote for

the contributions made by other experts in the team. The second module is a SOAP-based web service, which allows the server to share the updated KB with the client in PMML format (Data Mining Group., 2006). So, once the updated version of the KB has been downloaded from the server, the client can apply the mining algorithm offline. The teacher-client can also vote for a specific tuple.

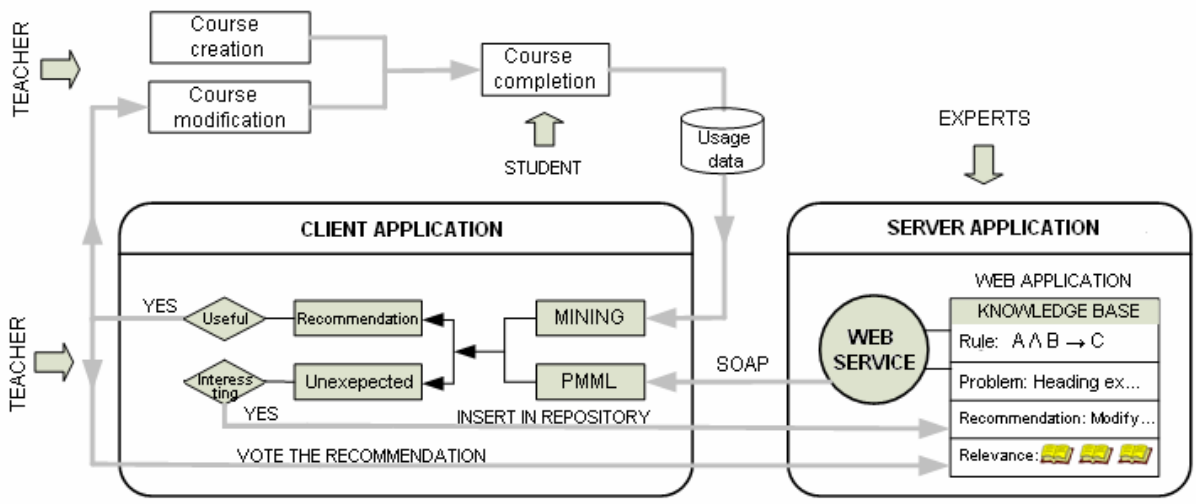

Figure 1. System architecture

The client application is part of the cyclical methodology (García, E., Romero, C. et al., 2006) that a teacher uses to construct a course. It is capable of detecting possible problems in the design and content of an e-learning course by adding a feedback or maintenance stage to the course.

There are several stages in this methodology: 1) the initial construction of a course; 2) the completion of the course by the students, during which usage information is transparently compiled and stored in a database; 3) the ongoing improvement stage, which coincides with the client application. This contains the nucleus of the rule mining algorithm proposed (section 3.2), which together with the KB expressed in PMML format classifies the rules found as: expected, if they coincide with the KB; or unexpected, if they do not. If the teacher applies a recommendation to the course, he or she is automatically voting for its usefulness in the knowledge database of the server. Unexpected tuples are ordered according to the IAS algorithm (Liu B., Wynne H., Shu C. Yiming M., 2000) and the teacher can tag any that s/he finds interesting. The experts then analyse these unexpected 'interesting' tuples and can choose to include them in the KB.

3.1. Weight-based rule evaluation measure

In order to help the teacher make decisions about which rules to apply, the rules and rule repository or knowledge database must be ordered in terms of interest. Therefore, an interestingness measure must be established based on the weights reflected by the following parameters:

1) Accuracy of the rules found by the current teacher according to the Predictive Apriori algorithm
2) How useful this rule has been to other teachers based on their votes.
3) How interesting the rule is according to a team of experts, also using a voting system.

Let U_1, U_2…, U_m, be m different teachers with different data-sources, S_i the set of expected association rules found by U_i (i=1,2,…m), $S = \{S_1, S_2,\ldots, S_m\}$; and let E_1, E_2…, E_m, be k different experts. According

to Good's definition of weight (Good I., 1950), the voting for rule *R* in *S* can be used to assign *R* a weight W_R. In practice, teachers are more interested in applying rules that have received greater support or more votes from other teachers.

Let $S = \{S_1, S_2 \ldots, S_m\}$ and $R_1, R_2, \ldots, R_n$ represents all the rules in *S*, then the weight of R_i can be defined as:

$$Wusers_{Ri} = \frac{NumVotesUsers(R_i)}{\sum_{j=1}^{m} NumVotesUsers(R_j)}$$

where *i*=1,2,…,n and *NumVotesUsers(R)* is the number of teachers that have voted for rule *R* in *S*.

By applying the same reasoning to the experts' votes:

$$Wexperts_{Ri} = \frac{NumVotesExperts(R_i)}{\sum_{j=1}^{k} NumVotesExperts(R_j)}$$

where *i*=1,2,…,n and *NumVotesExperts(R)* is the number of experts that have voted for rule *R* in *S*.

Therefore, the weight of rule R_i can be expressed as a weighted measure of the votes registered by the teachers and experts, so that:

$$W_{Ri} = Wusers_{Ri} * C_u + Wexperts_{Ri} * C_e \quad (1)$$

where C_u and C_e are the weighted coefficients with the opinions of the teachers and experts respectively, so that $C_u + C_e = 1$

Once the weight of each rule has been calculated, an interestingness measure can be devised, which we shall call weighted accuracy (*WAcc*) which includes the first factor mentioned at the start of this section: the predictive accuracy of the rule according to the Predictive Apriori (PA) algorithm.

Let $U_1, U_2, \ldots, U_m$, be *m* different teachers, then $WAcc_i$ can be defined for rule R_i obtained by the current teacher U_j (j=1,2…*m*) as:

$$WAcc_i = W_{R_i} * \frac{\sum_{j=1}^{m} acc(R_{ij})}{m}$$

where W_{Ri} is the weight of the rule according to equation (1), and $acc(R_j)$ (j=1,2,...*m*) is the predictive accuracy results returned by the PA algorithm for each teacher that has voted the rule R_i.

3.2. Distributed association rule algorithm

We have designed and implemented an association rule mining algorithm applied to education, which is based on the following algorithms: 1) Predictive Apriori for association rule discovery without parameters; and 2) IAS for subjective analysis and classification of unexpected rules by comparing them with a previously defined knowledge database about the field. The algorithm also includes the new weight-based interestingness measure discussed previously to recommend to the teacher any rules that: a) other teachers with a similar profile have found useful; or b) a team of validating experts has voted for in

terms of interest or validity. The algorithm implemented is especially useful in collaborative recommender systems, which can take advantage of the synergies offered by the network, in order to produce recommendations that are increasingly useful and precise.

The algorithm proposed is interactive and iterative. In each iteration, the teacher runs the mining algorithm in order to find the rules that will act as a basis for recommendations; the teacher can run this algorithm as many times as necessary. The system is made up of a general algorithm, shown in figure 2.

```
Input: Topic, level, difficulty: user profile;
       N: number of rules to discover
 1)  Num = N;
 2)  while (user don't stop) do
 3)     Rec, Rne = Rules_Mining_Algorithm (Num);
 4)     for each i-rule in Rec do
 5)        User_Vote_Recommendation(Reci);
 6)     end
 7)     for each i-rule in Rne do
 8)        if ( Interesting(Rne)) then
 9)           Add_to_KnowledgeBase(Rne);
10)        end if
11)     end
12)     Num += N;
13)  end while
14)  end all
```

Figure 2. Main algorithm

In step 1) the variable *Num* is initialised for the number of rules *N* that the teacher wishes to find; in step 2) a loop is begun and the instructions will be run until the teacher decides to stop it. Step 3) is the Mining sub-algorithm described in the next section, which returns the set of recommendations (*Rec*) and unexpected rules (R_{ne}) discovered. In steps 4) to 6), the teacher votes whether or not the recommendation has been useful, and in steps 7) to 11), he or she evaluates the unexpected rules to determine whether or not they are useful; unexpected rules might be added to the knowledge base (KB), subject to prior validation by the experts.

Let $U_1, U_2,\ldots, U_m$, be *m* different teachers, S_i the set of association rules found by U_k (k=1,2,…m); $S = \{S_1, S_2, \ldots, S_m\}$; and $R_1, R_2, \ldots, R_n$ all the rules in *S*; *acc*(R_i) (i=1,2,…n) is the predictive accuracy of R_i; *R* the set of rules discovered by the current teacher, R_e the set of expected rules, and R_{ne} the set of unexpected rules, then $R = R_e \; U \; R_{ne}$; KB is the set of rules that make up the knowledge database about the field. The proposed algorithm (Fig. 3) is designed as follows.

In step 1), the *GenRules* function discovers the association rules; this function is provided with the desired number of rules and calls on the Predictive Apriori algorithm described in section 2, which has been modified to include constraints on the items that might be present in the antecedent and consequent of the rules to be discovered. In step 2), the rules found are classified as expected, if they coincide syntactically with a rule in our knowledge database, or as unexpected if they do not. In steps 3) to 8), for

each rule $R_i \in R_e$, the new weight-based interestingness measure W*Acc* is calculated. In steps 9) to 12) the IAS algorithm is used to calculate the degree to which each unexpected rule R_{ne} coincides with the rules stored in the knowledge base (KB). In step 14) the set *Re* is ordered from highest to lowest based on the previously calculated W*Acc*. Step 15) displays all the recommendations corresponding to each of the previously ordered rules. Finally, in step 16), the teacher is given the chance to view the set of unexpected rules in order to assess which are interesting and possible candidates to be entered in the knowledge database.

Input: N: *number of rules to discover;*
Output: Rec: *recomendations;* R_{ne}: *unexpected rules*

1) R , acc(R) = GenRules(N); // *Predictive Apriori*
2) R_e, R_{ne} = Classify_Rules(R);
3) **for each** i-rule in R_e **do**
4) $$Wusers_{Re_i} = \frac{NumVotesUsers(R_{e_i})}{\sum_{j=1}^{m} NumVotesUsers(R_{e_j})};$$
5) $$Wexperts_{Re_i} = \frac{NumVotesExperts(R_{e_i})}{\sum_{j=1}^{k} NumVotesExperts(R_{e_j})};$$
6) $$W_{Re_i} = Wusers_{Re_i} * C_u + Wexperts_{Re_i} * C_e \; ; : C_u + C_e = 1$$
7) $$Wacc_{Re_i} = W_{Re_i} * \frac{\sum_{j=1}^{m} acc(R_{e_j})}{m};$$
8) **end**
9) **for each** i-rule in R_{ne} **do**
10) **for each** j-rule in KB **do**
11) $conform_{ij}$, $unexpConseq_{ij}$, $unexpCond_{ij}$, $bsUnexp_{ij}$ = **IAS( )**;
12) **end**
13) **end**
14) **High-Ranking** all the rules in R_e according to *Wacc*
15) **Output** the recommendations set Rec, corresponding to each rule in R_e
16) **Output** the unexpected rules R_{ne} according to IAS
17) **end all**

Figure 3. Rules Mining Algorithm

4. Implementation of the system

The client application and server make up the hybrid recommender system based on KBS and CFS, where recommendations for improvements to the course are made on the basis of the knowledge databases created and managed in the server according to the different teacher profiles. Furthermore, collaborative filtering is used as a complementary approach, which will filter and organise the priority of

recommendations depending on the votes registered by experts and teachers with a similar profile. The experts explicitly vote for tuples by indicating different degrees of preference on a form in the web application; however, the teachers vote implicitly in order to avoid one of the main problems of CFS. This problem relates to how to encourage users to vote or evaluate. In this case, if teachers apply one of the recommendations to their course, they are automatically voting for to the applicability of the tuple.

4.1. Client application

The client application and the Predictive Apriori algorithm have been implemented in Java language due to its multi-platform characteristics. One the main features of this application is its specialisation in educational environments, using domain specific attributes, filters and restrictions on the dataset for the usage of the e-learning course to be analysed.

The application has four basic panels:

Pre-processing. In this panel, the teacher first of all selects the origin of the data for mining. The format of the main input data is a Moodle MySQL database. Once the data have been selected, the programme displays all the numerical attributes present. In order to make the rules discovered easier to understand and significantly reduce the run time of the search algorithm, these attributes must be discretized.

Configuration parameters. This module displays the parameters used by the Predictive Apriori mining algorithm, including the number of rules to be discovered, as well as a series of restrictions that the teacher can enter in relation to the maximum number of items present in the antecedent or consequent of the rules discovered. If the teacher does not wish to change the set-up, the default parameters can be used.

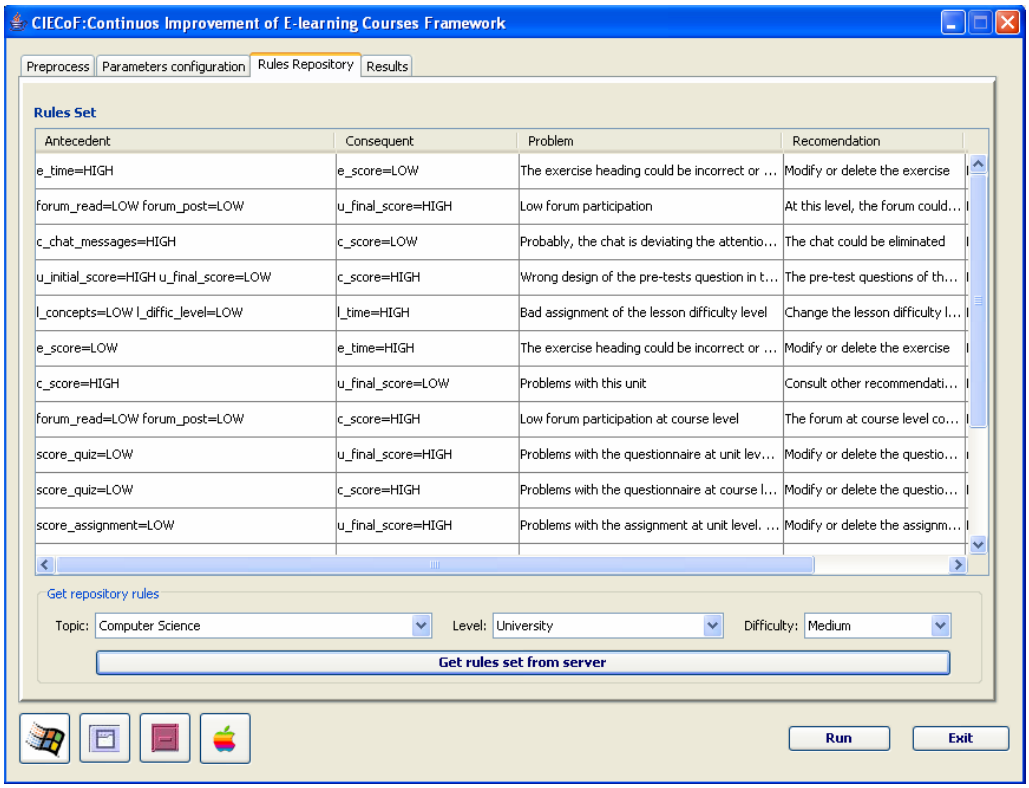

Figure 4. Client interface

Rules Repository. The rules repository, shown in Figure 4, is the knowledge database, on the basis of which subjective analysis of the rules discovered is performed. Before running the algorithm, the teacher downloads the relevant knowledge database from the server, in PMML format, in accordance with his/her profile. The personalisation of the items returned is based on three filtering parameters, associated with the type of course to be analysed: 1) the area of knowledge; 2) level of education; and 3) the difficulty of the course. The fields included in the returned repository are: the rule itself, the problem detected by the rule and a possible recommendation for its solution. In order to identify each tuple, additional data are also included, such as the author, date and evaluation of the rule. The rules repository is created on the server (section 4.2), based on the educational considerations of the experts and the experience gained on other similar e-learning courses.

Results. Once the application parameters have been configured, or using default values, the teacher runs the algorithm. This panel shows the results obtained in a table, with the following fields:

Rule-Problem-Recommendation-Score-APPLY

The recommendation can be one of two types:

1) Active, if it implies a direct modification of the course content or structure. Active recommendations can be linked with: modifications in the formulation of the questions or the practical exercises/tasks assigned to the students; changes in previously assigned parameters such as course duration or the level of difficulty of a lesson; or the elimination of a resource such as a forum or a chat room.

2) Passive, if they detect a more general problem and point the teacher towards more specific recommendations.

For active recommendations, by clicking the *APPLY* button, the teacher will be shown the area of the course the recommendation refers to. If it is an active recommendation and the teacher applies it, s/he is implicitly voting for that recommendation.

4.2. Server application

For the server, a web application (see Figure 5) was implemented to manage the knowledge database, using JSP (Java Server Pages). For complete access to all the editing options for the repository, a basic profile was created, which is the profile of the education expert, who has permission to enter new rules in the repository and vote for existing ones. Based on the votes registered by experts, the parameter $W_{experts}$ is calculated. Implicit votes are also stored, which are registered by clients in their local analyses, and based on these votes, $W_{teachers}$ is calculated.

In order to allow information to be exchanged between client and server, a web service was implemented to manage the exchange of the PMML archive, which contains the repository. With every exchange, the parameters used in the algorithm described previously are recalculated and the tuples are reordered in the repository, taking into account the accuracy parameter W*Acc*.

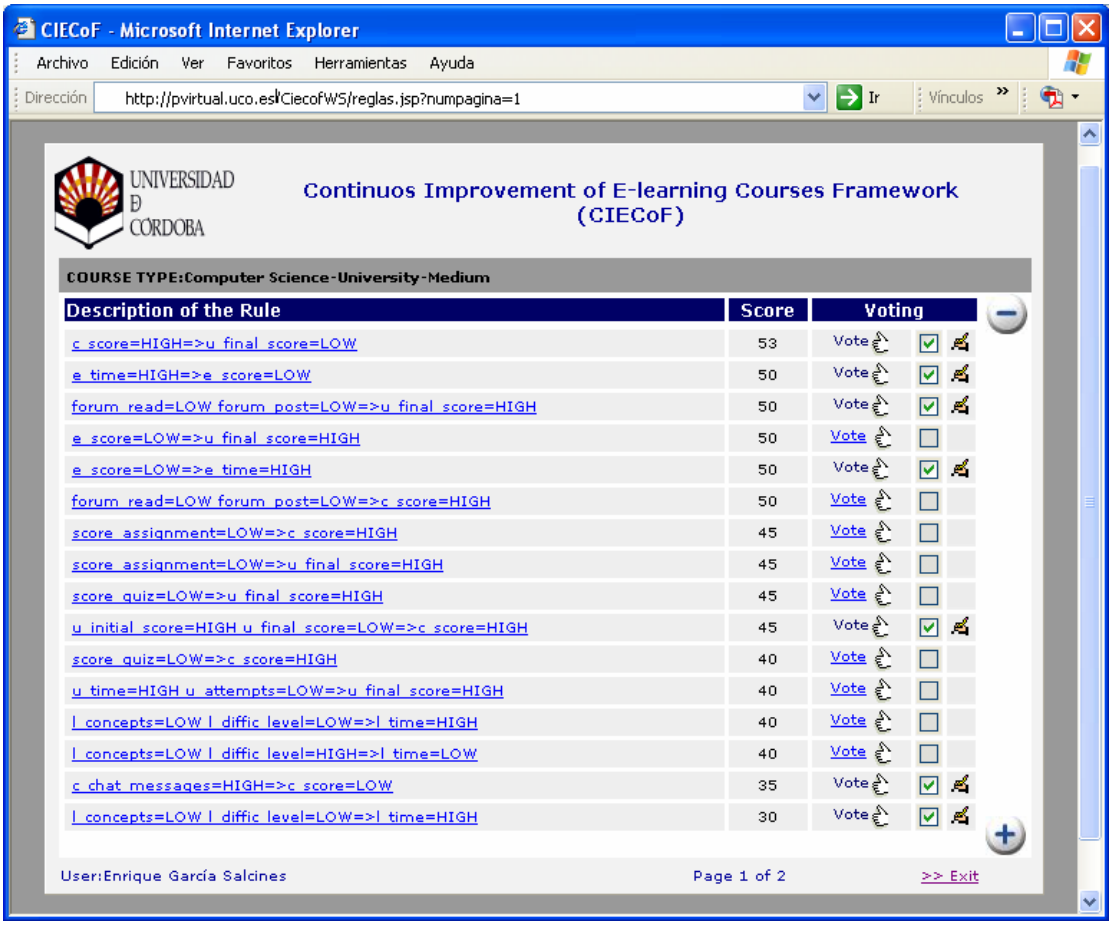

Figure 5. Sever application interface

5. Experimental Results

In order to demonstrate the architecture, the mining process had to be applied to a dataset. In 2004-2005, a pilot experiment was carried out in Cordoba (Spain) in relation to the technological literary of women in rural settings, called "*Cordobesas Enredadas*". In this project, 7 adaptive web-based courses were developed, based on subjects included in the ECDL *(European Computer Driving Licence)* using the Linux Operating System (Guadalinex distribution) and the free distribution office package Open Office.

The courses were developed using the authoring tool INDESAHC (De Castro C., García E., Romero C., Ventura S., 2004), which can be used to create hypermedia adaptive courses that are compatible with Moodle..

5.1. INDESAHC data

The definition of the course syllabus in INDESAHC was based on a hierarchical domain model made up of teaching units divided into lessons, each of which contained a series of concepts to explain or assess through scenarios or web pages (see Figure 6). An adaptation model was also included in order to adapt the contents to each student's level of knowledge. For this, the Adaptive Link Hiding and Annotation (De Bra, P. & Calvi, L., 1998) technique was used, once the course contents had been classified according to different levels of difficulty

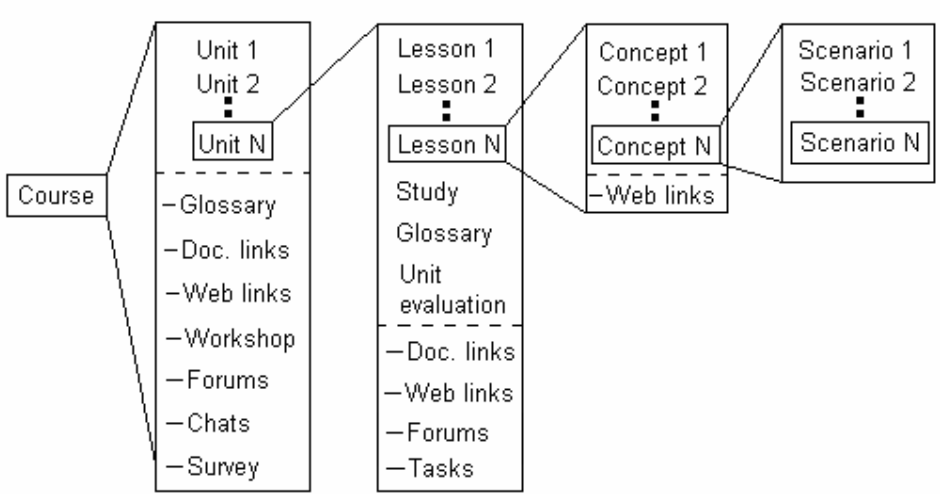

Figure 6. INDESAHC domain model

Table I shows, on one hand, the data attributes related with the adaptive hypermedia courses which are added as new tables to Moodle database. On the other hand, we can see attributes related to other teaching resources such as forums, chat rooms, questionnaires and tasks which are also introduced from the INDESAHC interface. Once the course has been generated and uploaded using Moodle, these resources are automatically inserted in accordance with the template used in the different sections together with the adaptive hypermedia course.

Level	Attribute	Description
Course	duration	Estimated course duration according to the teacher
	c_time	Time taken by the student to complete the course
	c_score	Average final score for the course
	c_attempt	Number of attempts before passing the course
	c_quiz_attempt	Total number of attempts in the quiz
	c_quiz_time	Total time taken in the quiz
	c_quiz_score	Score obtained in the quiz
	c_chat_messages	Number of messages sent in the chat room
	assignment_score	Score in the assignment
	forum_read	Number of messages read in the forum
	forum_post	Number of messages posted in the forum
	doc_view	If the document or web link has been viewed
Unit	u-lessons	Number of lessons in a unit
	u_time	Time taken by the student to complete the learning unit
	u_initial_score	Student's score in the unit pre-test
	u_final_score	Student's final score on completing the unit
	u_attempt	Number of attempts before passing the unit
	forum_read	Number of messages read in the forum
	forum_post	Number of messages posted in the forum
	assignment_score	Score in the assignment
	doc_view	If the document or web link has been viewed
Lesson	l_concepts	Number of concepts in the lesson
	l_time	Time taken by the student to complete the lesson
	l_diffic_level	Level of difficulty of the lesson as defined by the teacher
Exercise	e_time	Time taken by the student to complete the exercise
	e_score	Score obtained in the exercise

Table I. *Attributes used in association rules mining process*

5.2. Data pre-processing

Before applying association rule mining, the input data must first be pre-processed in order to adapt them to our data model. This pre-processing stage includes a series of stages such as data cleaning, the transformation of continuous variables into discreet variables and the integration of data when they are from different sources. In this case, data cleaning was carried out for two very common reasons: firstly, it was discovered that very high values were often recorded for the attribute *time* because the student had left the computer without first exiting the exercise, concept or section; in order to correct this, any times that exceeded a maximum established value were considered noisy data, and this maximum value was assigned to any apparently erroneous data. Secondly, it was discovered that some students had not completed all the course activities. Whenever possible, the students were contacted and asked to complete the course so that their information can be used. When this was not possible, the information regarding said student was discarded.

Once the data were selected, the programme displayed all the numerical attributes present. In order to make the rules discovered easier to understand and significantly cut down on the run time of the search algorithm, these attributes had to be discretized. The transformation into discreet variables can be seen as a categorisation of attributes that takes a small set of values. The basic idea (H. Liu, F. Hussain, C.L. Tan, and M. Dash., 2002) involves partitioning the values of continuous attributes within a small list of intervals. Each resulting interval is an estimation of a discreet value of the attribute. Our process of discretization used three possible nominal values - LOW, MEDIUM and HIGH - and three methods: equal width method, equal frequency method and a manual method, where the teacher sets the limits of the categories manually.

Normally, when working on a data mining problem, a single dataset must first be established from all the data that come from different sources. In this case, there were two sources: 1) the tables that stored student monitoring data using the specific attributes of INDESAHC; and 2) the tables used by Moodle, which stored information about the use of other teaching resources on the course such as forums, chat rooms and tasks. Using these data, a temporary database was created to which rule mining was applied. MySQL was used to carry out these tests because our default database server is Moodle.

Before applying the rule mining algorithm, the teacher could also restrict the search domain, by specifying the level of granularity of the analysis, for example at a subject, lesson or exercise level. The resulting temporary table in this case would, therefore, only contain attributes and transactions from students relating to the selected level. The system could also find interesting relationships between attributes from different tables, for example if the teacher selected a course-subject or subject-exercises, the temporary table created would contain attributes and transactions from more than one table.

5.3. Analysis of the recommendations effectiveness

In order to verify the effectiveness of the changes made by the teachers based on the recommendations suggested by the system, it is important to bear in mind two points of view: 1) the teacher's perspective, in terms of the percentage of apparently corrected problems - based on initial recommendations - that reappear in successive courses with different groups of students; and 2) the perspective of the students in relation to how the removal of problems based on the recommendations influences their final score. Two starting hypotheses can be derived from these aspects: firstly, if the changes made by the teacher are 100% effective, these problems should not be detected again in subsequent groups of students; and secondly, if these problems do not reoccur, students' scores should improve.

In an ongoing cycle of course improvement, such as the one proposed here, with successive corrections based on the usage data of different groups of students, $TotalRec_1$ represents the total number of

recommendations found when the usage data of the first group of students were analysed, which led to changes in the structure or content of the course; if $TotalRec_{1,i}$ is the total number of recommendations that are repeated in consecutive courses with other groups of students, the effectiveness of the changes made can be calculated, based on the recommendations proposed in the initial stage (the first course run) in relation to stage i ($i=2,3...N$) , which corresponds to subsequent courses, as follows:

$$EfectRec_1 = \frac{TotalRec_1 - TotalRec_{1,i}}{TotalRec_1} \qquad (2)$$

We can measure the effectiveness of the corrections made in terms of the students, by comparing the average score and standard deviation in subsequent courses.

In order to calculate (2) and compare the students' scores, the basic material used was the course on "Spreadsheets" and two groups of 45 students who completed consecutive courses. In order to eliminate the influence of external factors such as previous computing knowledge, average age of the group and level of education, which might alter the results of the research, the composition of the two groups was forced so that they met the following requirements: 1) the students must have no prior knowledge of computers, which was relatively easy since the courses were aimed at computer literacy in rural settings; 2) the average age of the group had to be the same; 3) the level of education could not be above intermediate.

Table II shows the results from the teacher's point of view when applying our system consecutively to data from the two groups of students. The column "*New*" refers to the initial recommendations provided by the system to problems detected in the course and which the teacher considered useful and applicable; the column "*Rep*" refers to initial recommendations that, even though the teacher applied them, reappeared in the same tuples in consecutive courses. Table III shows the results from the point of view of the student. Column "*NRep*" refers to the tuples that did not reappear; it also shows the average final scores and standard deviation of each group and calculates the p values comparing group 1 and group 2.

Group	New	Rep	Total	EfectRecom (%)
1	21	0	21	0
2	5	6	11	72,7

Table II: *Results from the point of view of the teacher*

Group	NRep	Score	p-value 1-2
1	0	6,55 +0,30	< 0,0001
2	15	6,95 +0,56	

Table III: *Results from the point of view of the student*

Analysis of the data in Table II and III points to several conclusions:

1) As anticipated in the initial hypothesis, the effectiveness percentage gets closer to 100% with subsequent editions of the course. Problems that reoccurred were because of changes made to the course design, which had a strong subjective component, for example changing the difficulty level of a lesson or the estimated duration of a subject.

2) In each group, new problems were detected, which had not been spotted previously, and therefore new recommendations were made to resolve them. The reason for this might be that, in spite of attempts to equal out the composition of each group, we were working with people and highly subjective characteristics such as intellect and skill.

3) Not only did the effectiveness percentage increase, but the total number of recommendations linked with the detected problems also decreased, which is another sign that the course was continually improving.

4) When the scores achieved by both groups were compared, a slight improvement was observed, which also indicates the effectiveness of the system proposed.

5.4. Description of the discovered rules

The results described below are from trials carried out for the "Word Processing" course. Below is a description of a couple of expected rules discovered, in other words rule that coincide with the knowledge database.

1) **IF** (e_time [25] = HIGH) **THEN** (e_score[25] = LOW), accur. = 0.85

This rule means that if the student took a long time to complete the exercise, she would get a low score. This rule discovered that there was a problem with this specific exercise, which was part of the subject "application use", lesson "first steps with the word processor" and concept "renaming and saving a document". The exercise was an INDESAHC interactive video scenario in which the student had to simulate the necessary steps to complete an activity using the mouse. In this specific case, it was confirmed that the question was ambiguous and could be interpreted in several ways, and so the wording was changed. Other rules with a similar format were found, but related to multiple-choice or linking type questions.

2) **IF** (u_forum_read [2] = LOW) and (u_forum_post [2] = LOW) **THEN** (u_final_score [1] = HIGH), accur. = 0.75

This rule means that if the student did not send or read many messages in forum 2 (unit 1), then she would get a high score for this unit. This rule revealed that the forum was not necessary or that there were problems with the tutor. This type of rule questions the need for a forum at certain levels of the domain hierarchy; in fact in this case, the forum was removed.

It should also be mentioned that many rules were discovered, but which did not provide any useful information to detect problems or they give obvious information for the teacher. An example of this type of rules is the next:

3) **IF** (assignment_score [9] = HIGH) **THEN** (u_final_score[2] = HIGH), accur.= 0.72

This rule shows that if the score of the assignment 9 is high then the obtained final score of unit 2 is high. This rule is totally logical for the teacher and it do not give any new information about how to improve the course.

6. Conclusions and Future Work

This paper presents a collaborative recommender system that uses distributed data mining in order to continuously improve e-learning courses. This system allows teachers with a similar profile to share their research results as a consequence of applying mining locally on their own courses. Furthermore a new interactive and iterative association rule mining algorithm was developed, using a new weight-based evaluation measure for the rules discovered and taking into account the opinion of experts and the

teachers themselves in order to produce increasingly effective recommendations. Experimental trials were performed taking into account two points of view: that of the teacher making the changes based on the recommendations provided by the system; and that of the student doing the course, after it had been modified by the teacher. The final results demonstrate our starting hypotheses: fewer problems are detected in subsequent courses and the students' final scores improve as the teacher corrects the problems. Even though the scores improved slightly, this area deserves greater attention, and we aim to conduct a more detailed study with more groups in order to find more significant differences. Research is also being carried out with more students and teachers in order to discover which other attributes of the data model - in addition to the final score - could be taken into account for comparison.

Acknowledgement

Part of this work was supported by the Provincial Government of Cordoba, under the project reference *ECDL/DIPUCO/MEM/04-0001bis*.

References

Agrawal, R., et al.:1996, 'Fast discovery of association rules', *In Advances in Knowledge Discovery and Data Mining, Menlo Park*, CA: AAAI Press, pp. 307-328.

Barnett, V.; Lewis, T.:1994; *Outliers in Statistical Data*. John Wiley & Sons.

Brusilovsky, P.:1996, 'Methods and techniques of adaptive hypermedia', *User Modeling and User-Adapted Interaction*, 6(2-3), pp. 87-129

Brusilovsky, P.: 2003, 'Adaptive and Intelligent Web-based Educational Systems', *International Journal of Artificial Intelligence in Education*, pp. 156-169.

Brusilovsky, P., Schwarz, E., & Weber, G.:1996, 'ELM-ART: An intelligent tutoring system on World Wide Web', *Third International Conference on Intelligent Tutoring Systems*. Berlin: Springer Verlag, Vol. 1086, pp. 261-269.

Burke, R.:2000a, 'Semantic ratings and heuristic similarity for collaborative filtering', *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, Texas, July 30th–August 3rd.

Burke, R.:2000b, 'Knowledge-based Recomendador Systems'. *In A. Kent (ed.), Encyclopedia of Library and Information Systems*. Vol. 69, Supplement 32. New York: Marcel Dekker.

Chen W. & Wasson B.:2002, 'Coordinating Collaborative Knowledge Building'. *Proc of International Conference Applied informatics*.

Cheung D. W., Ng V. T., Fu A.W., Fu Y.:1996, 'Efficient mining of association rules in distributed databases', *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 911–922.

Costaguta R.:2006, 'Una Revisión de Desarrollos Inteligentes para Aprendizaje Colaborativo Soportado por Computadora'. *Revista Ingeniería Informática*, Nº13, http://www.inf.udec.cl/revista.

Data Mining Group.:2006, Predictive Model Markup Language (PMML). Available at http://www.dmg.org/pmml-v3-0.html.

De Bra, P. & Calvi, L.:1998, 'AHA! An open Adaptive Hipermedia Architecture', *The New Review of Hipermedia and Multimedia*, 4, pp. 115-139.

De Castro C., García E., Romero C., Ventura S.:2004, 'Herramienta autor INDESAHC para la creación de cursos hipermedia adaptativos'. *Revista latinoamericana de tecnología educativa*. Vol. 3, 1, pp. 349-367.

Eliassi-Rad, T. and Shavlik, J.:2003, '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, pp. 35-88.

Freyberger, J., Heffernan, N., & Ruiz, C.:2004, 'Using association rules to guide a search for best fitting transfer models of student learning'. In *Workshop on analyzing student–tutor interactions logs to improve educational outcomes at ITS conference*. Maceio, Brazil, pp. 1-4.

García, E., Romero, C. et al.:2006, '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, pp. 887-895.

Gaudioso E., Santos O., Rodriguez A., y Boticario J.:2003, 'A Proposal for Modeling a Collaborative Task in a Web-Based Collaborative Learning Environment', en Proc. *9th. International Conference on User Modeling*. USA.

Geyer-Schulz, A. et al.:2003, 'An Architecture for Behavior-Based Library Recomendador Systems', *Information Technology and Libraries*. 22(4), pp.165-174.

Good I.:1950, *Probability and the weighting of evidence*. Charles Griffin, London, 1950.

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

Heift, T., & Nicholson, D.:2001, 'Web delivery of adaptive and interactive language tutoring', *International Journal of Artificial Intelligence in Education*, 12(4), pp. 310-324.

Itmazi, J.A.S.:2005, 'Sistema Flexible de gestión del e-learning para soportar el aprendizaje en las universidades tradicionales y abiertas'. PhD Thesis. University of Granada, Spain.

Li, J., & Zaïane, O.R.:2004, 'Combining usage, content, and structure data to improve web site recommendation'. In *Proceedings of the International conference on e-commerce and web technologies*, Zaragoza, Spain, Springer, Lecture Notes on Computer Science, 3182, pp. 305–315.

Liu B., Wynne H., Shu C. Yiming M.:2000, 'Analyzing the Subjective Interestingness of Association Rules', *IEEE Inteligent System*.

Lu, J.:2004, 'Personalized e-learning material recommender system'. In *Proceedings of the International conference on information technology for application*, London, England. pp. 374–379.

Markellou, P., Mousourouli, I., Spiros, S., & Tsakalidis, A. :2005, 'Using semantic web mining technologies for personalized e-learning experiences'. In *Proceedings of the web-based education*, Grindelwald, Switzerland, pp. 461–826.

Merceron, A. & Yacef, K.:2004, 'Mining student data captured from a web-based tutoring tool: Initial exploration and results'. *Journal of Interactive Learning Research, 15(4)*, pp. 319–346.

Minaei-Bidgoli, B., Tan, P., & Punch, W.:2004, 'Mining interesting contrast rules for a web-based educational system'. In *Proceedings of the International conference on machine learning applications*, Louisville, KY. pp. 1-8.

Romero, C., Ventura, S., Bra, P. D., & de Castro, C.:2003, 'Discovering prediction rules in AHA! Courses', *En 9th International User Modeling Conference* (Vol. 2702), Berlin: Springer Verlag, pp. 25-34

Romero, C., Ventura, S., & Bra, P. D.:2004, 'Knowledge discovery with genetic programming for providing feedback to courseware author'. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 14(5), pp. 425–464.

Romero, C., Ventura, S.: 2006, 'Educational data mining: a survey from 1995 to 2005'. *Expert Systems with Applications*, 33(1), pp. 135-146.

Rosta F., Brusilovsky, P.:2006, 'Social navigation support in a course recommendation system', *Adaptive Hypermedia and Adaptive Web-Based Systems: 4th International Conference, AH 2006*, pp. 91-100.

Savasere A., Omiecinski E., and Navathe S. B.:1995, 'An efficient algorithm for mining association rules in large databases', *in Proceedings of 21st International Conference on Very Large Data Bases*. VLDB, pp. 432–444. [Online]. Available at: http://www.vldb.org/dblp/db/conf/vldb/

Scheuermann P.:2001, 'Distributed web log mining using maximal large itemsets', *Knowledge and Information Systems*, vol. 3, no. 4, pp. 389–404.

Silberschatz, A., Tuzhilin, A.:1996, 'What makes pattterns interesting in Knoledge discovery systems', *IEEE Trans. on Knowledge and Data Engineering*. 8(6), pp.970-974.

Skillicorn D. B. and Wang Y.:2001, 'Parallel and sequential algorithms for data mining using inductive logic', *Knowledge and Information Systems*, vol. 3, no. 4, pp. 405–421.

Srivastava, J.; Mobasher, B.; Cooley, R.:2000, 'Automatic Personalization Based on Web Usage Mining', *Communications of the Association of Computing Machinery*, pp. 142-151.

Tan P., Kumar V.:2000, 'Interesting Measures for Association Patterns: A Perspectiva', Technical Report TR00-036. Department of Computer Science. University of Minnnesota.

Tang T., McCalla, G.:2005, 'Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment', *International Journal on E-Learning*, 4(1), pp. 105-129.

Terveen, L. and Hill, W.:2001, '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, pp. 487-509.

Tobias S.:2001, 'Finding Association Rules That Trade Support Optimally against Confidence', *Lecture Notes in Computer Science*, Vol. 2168, pp. 424+.

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

Wang, F.:2002, 'On using data-mining technology for browsing log file analysis in asynchronous learning environment'. In *Proceedings of the Conference on educational multimedia, hypermedia and telecommunication*, Denver, Colorado. pp. 2005– 2006.

Zaiane O.:2002, 'Building a Recommender Agent for e-Learning Systems', *Proc. of the International Conference on Computer in Education*, pp. 55-59.

Zaïane, O., & Luo, J.: 2001, 'Web usage mining for a better web-based learning environment'. In *Proceedings of conference on advanced technology for education*, Banff, Alberta, pp. 60–64.

Zan, H. et al.:2004, 'A graph model for E-commerce Recomendador systems', *Journal of the American Society of Information Science and Technology*, 55(3), pp.259-274.

Zhang C., Zhang S.:2002, *Association Rule Mining*. Berlin: Springer, chapter 7.

Zheng Z., et al.:2001, 'Real world performance of association rules'. *In Proceedings of the Sixth ACM-SIGKDD*.