A Web Mining Methodology for Personalized Recommendations in E-commerce

M.N. Moreno*, F.J. García, V. López and M.J. Polo

Dept. Informática y Automática. University of Salamanca. Plaza Merced s/n. 37007 Salamanca. Spain
*E-mail: mmg@usal.es

Abstract. A current trend in the e-commerce systems is to incorporate mechanisms for personalized product recommendation in order to obtain new customers and retain existing ones. In this paper, a recomender methodology based on web data mining is proposed. The work deals with the problem of introducing new products and new customers which do not have a profile. We use the available information from other users to discover patterns in data through association rule algorithms. We suggest the use of intelligent agents for its implementation. The core agent generates and refines association rules by means of a characteristic algorithm which contributes to avoid the well-known problems of the collaborative filtering techniques and leads to the development of more efficient recommendations.

1 Introduction

The electronic business activities have suffered a quick growth in the last years. Consequently, the amount of available information is greater than a consumer can manage. E-commerce systems need to provide users with mechanisms for selective retrieval of web information. On the other hand, this growth has contributed to increase the competition among the business organizations. A way for improving its competitiveness is to take advantage of business intelligence techniques as data mining and intelligent agents.

Data mining techniques provides the objective information needed to make decisions about business. There are two general categories of data mining problems: *prediction* and *knowledge discovery*. Prediction problems are described in terms of specific goals, which are related to past records with known answers. These are used to project to new cases. Knowledge discovery problems usually describe a stage prior to prediction, where information is insufficient for prediction, the goal is to discover rules and segments of the data that behave similarly [17]. Data mining problems are resolved by employing *supervised* and *unsupervised* algorithms. In the first case a *learning* phase is necessary to build a predictive model. Unsupervised algorithms are used in Knowledge discovery modeling. This is a descriptive task.

The application areas of the data mining in commercial systems are multiple. Market management is one of the best known. Data mining algorithms can find consumers profiles in the corporate databases that can be used to drive promotional campaigns and to do a more effective marketing. Algorithms to search sequential patterns can show the behavior of the consumers such as the sequence in which they purchase products or services. Other application areas are customer retention, competitive analysis and fraud detection [1].

In the electronic business environment, a way to obtain new customers and retain existing ones is the personalized product recommendation. E-commerce applications that incorporate recommender systems provide users with intelligent mechanisms to search products to purchase. This is a way to avoid the problem of *information overload* due to the great quantity of information accessible through the Web [2].

The two main recommendation methods are: collaborative filtering and a content-based approach [6]. The first technique is one of the most successful methods and it was based initially on nearest neighbor algorithms. These algorithms predict product preferences for a user based on the opinions of other users. The opinions can be obtained explicitly from the users as a rating score or by using some implicit measures from purchase records as timing logs [14]. In the content based approach text documents are recommended by comparing between their contents and user profiles. The weights for the words extracted from the document are added to the weights for the corresponding words in the user profile, if the user is interested in the page [6]. The main shortcoming of this approach in the e-commerce application domain, is the lack of mechanisms to manage web objects such as motion pictures, images, music, etc. Besides, it is very difficult to handle the big number of attributes obtained from the product contents. Collaborative filtering also has limitations in the e-commerce environment. Rating schemes can only be applied to

homogeneous domain information. Besides, sparsity and scalability are serious weaknesses which would lead to poor recommendations [3]. Sparsity is due to the number of ratings needed for prediction is greater than the number of the ratings obtained because usually collaborative filtering requires user explicit expression of personal preferences for products. The second limitation is related to performance problems in the search for neighbors. These problems are caused by the necessity of processing large amount of information. The computer time grows linearly with both the number of customers and the number of products in the site.

The quality of the recommendations for the users has an important effect on the clients' retention. Users refuse poor recommender systems which can cause two types of error: *false negatives*, which are products that are not recommended, though the customer would like them, and *false positives*, which are products that are recommended, though the customer does not like them [3]. The most serious errors are false positives, because these errors will cause negative reactions in the customers and thus they won't probably visit the site again. The use of data mining algorithms, like the one proposed in this work, to find customers characteristics that increase the probability of buying recommended products, can help to avoid these problems.

The process of applying data mining techniques on web data to obtain customer usage patterns is known as web mining. These methods build models based mainly on users' behaviour more than in subjective valuations (ratings). This is the main advantage of this approach that allows avoiding the problems associated with traditional collaborative filtering techniques [8]. Patterns extracted from web data can be applied to web personalization applications.

In this work, we present a recommendation methodology based on web mining that uses diverse information as user's attributes, rating and usage data. The core of the methodology is an unsupervised data mining algorithm which generates and refines association rules in order to discover knowledge for making personalized recommendation. The goal in *knowledge discovery* modeling is to discover rules and segments of the data that behave similarly. We applied an association rule algorithm to discover patterns in data. However, most of the existing algorithms have the drawbacks that they discover too many patterns which are either obvious or irrelevant and, sometimes, contradictions, between rules appear. We propose a refinement method in order to obtain stronger rules that reinforce the relation between items. The process to refine association rules is based on the concept of *unexpected patterns*. Our proposal should provide recommender systems with more relevant patterns that minimize the recommendation errors. The architecture suggested for these systems is constituted by intelligent agents; one of them is in charge of doing data mining tasks.

2 Related Work

In the last years many recommender systems have been developed. Systems based on the content approach present the drawbacks commented previously. The collaborative filtering method is more suitable for systems which manage multimedia objects. The GroupLens research system [5], Ringo [14] and Video Recommender [4] are three examples of systems based on this approach. The usual technique used in these systems is based on correlation coefficients. The method requires user ratings about different recommendable objects. Correlation coefficients showing similarities between users are computed from the ratings. Then, recommendations based on these coefficients can be made. The procedure presents the sparsity problem. Another obstacle is the first-rater problem that takes place when new products are introduced [5].

There are two approaches for collaborative filtering, *memory-based* (*user-based)* and *model-based* (*item-based)* algorithms. ***Memory-based*** algorithms, also known as *nearest-neighbor* methods, were the earliest used [13]. They treat all user items by means of statistical techniques in order to find users with similar preferences (*neighbors*). The prediction of preferences (recommendation) for the active user is based on the neighborhood features. A weighted average of the product ratings of the nearest neighbors is taken for this purpose. The advantage of these algorithms is the quick incorporation of the most recent information, but they have the inconvenience that the search for neighbors in large databases is slow [15].

Data mining technologies, such as Bayesian networks, clustering and association rules, have also been applied to recommender systems. ***Model-based*** collaborative filtering algorithms use these methods in the development of a model of user ratings. This recent approach was introduced to reduce the sparsity problem and to get better recommender systems.

The Bayesian network analysis is a classification technique that formulates a probabilistic model for collaborative filtering problem. The underlying structure used for classification is a decision tree representing user information. The predictive model is built off-line by a machine learning process and used after to do recommendations to the active users. Te process is fast and simple and this is very

suitable for systems in which consumer preferences change slowly with respect to the time needed to built the model [15].d

In a recent work [2], the authors propose the use of methods from both categories in order to avoid the commented problems. The support vector machine (SVM) memory-based technique is used for content-based recommendations, and the latent class model (LCM), that is a model-based approach, is used for collaborative recommendation. The last technique considers a predefined number of "latent" classes (Z) corresponding to preference patterns. The probability that customer x buys product y is computed by means of the Bayes rule from previous calculation of the z class prior probability and the individual (x and y) conditional probabilities regarding z. The inconvenience of this method is the way to solve the problem of introducing new users. In this case the recommendations are based on the averaged opinions of the remainder users. The best rated products are recommended without consider characteristic attributes of the user.

Rule-based approaches have also been applied to overcome problems that personalized systems have [6]. The data should be processed before generating the rules.

In [3] a recommendation methodology that combines both data mining techniques is proposed. First a decision tree induction technique is used in the selection of target customers. Later, association rules are generated and used for discovering associations between products.

Clustering techniques identify groups of users who appear to have similar preferences. Predictions are based on the user participation degree in the clusters. These techniques are not very accurate and they can be useful in a preliminary exploration of the data.

A graphical technique that has yielded better results than nearest neighbors is horting [18]. Nodes in the graph represent users and edges between nodes indicate degree of similarity between users. Prediction are produced by walking the graph to nearby nodes and combining the opinions of the nearby users.

The algorithm of nearest neighbors has also been applied in combination with data mining techniques. Lee et al. [6] create a user profile that is effective within a specific domain by using a nearest neighborhood-based method. They expand the traditional method to find profiles valid over the all domains. For each user, neighbors' transaction information is used to generate web object association rules.

Our proposal is a different model-based approach that deals with the case of new users in which an initial user profile is not available. We apply an association rule algorithm in order to find initial patterns. Rating and users' information is used in the rule generation procedure. We use other users' attributes to refine the rules and obtain specific rules that consider the particularities of each user. The refinement procedure is based on the concept of unexpectedness.

3 Recommendation methodology

Nowadays, many commercial recommender systems use customer's preference ratings for product recommendation. In most cases, the best rated products are recommended without consider characteristic attributes of the user.

In this section, a methodology for personalized recommendations is presented. The main objective is to provide a simple procedure which avoids the problems commented previously (figure 1). It uses information about consumer preferences and user attributes.

The first step of the methodology is the selection of the best rated products. This is a data mining task consisting of evaluating rating information of a product in relation to the products of the entire transactions' database. A list of products ordered from most to less popular is generated. It is necessary to establish a minimum threshold value of the ratings for selecting the best products. This value depends on several factors (number of available products, the scale and the measure units…). The aim of this step is to reduce the number of association rules generated and to obtain rules applicable to a wide range of customers.

The selected records are the input for the second step in which the association rules are generated. The rules relate product attributes with user attributes and preferences, in this way it is possible to identify products that will be of interest to a given customer with a specific profile. The initial rules are refined by means of an algorithm described in the section 5.3. The aim of the refinement procedure is to obtain strong patterns that avoid the false positive recommendations. The association rules are also a solution to the problem of the introduction of new users. When a new customer accedes to the system, he is checked to obtain his profile and to generate recommendations for him.

In the third step, the recommendations based on previous steps are made. The recommendations are based on the patterns obtained which relate user attributes with product attributes . For new and old users the system recommends products with characteristics adapted to their profile. The rules enable to

recommend new products whose characteristics agree with the preferences of the users. New products with new characteristics are always recommended. This is the way to deal with the first-rater problem.

The process is iterative and uses the new information about products and users to feedback the system. New association models are built when the system has a significant quantity of new information.

The implementation of the system is carried out by using three main agents which take charge of interacting with the user, managing the information and generating the recommender models.

- Data mining agent: this agent uses the data mining algorithms in order to generate the models used in the recommendations. It is in charge of generating and refining association rules. This agent uses the information provided by the data management agent periodically.
- Recommendation agent: the agent receives the requests from the users, takes their preference profile and uses the data mining models to make the personalized recommendations.
- Data management agent: It collects and manages the storage of the information about new user preferences and new products. It is connected with the data mining agent to provide the data that are used periodically in the generation of new models.

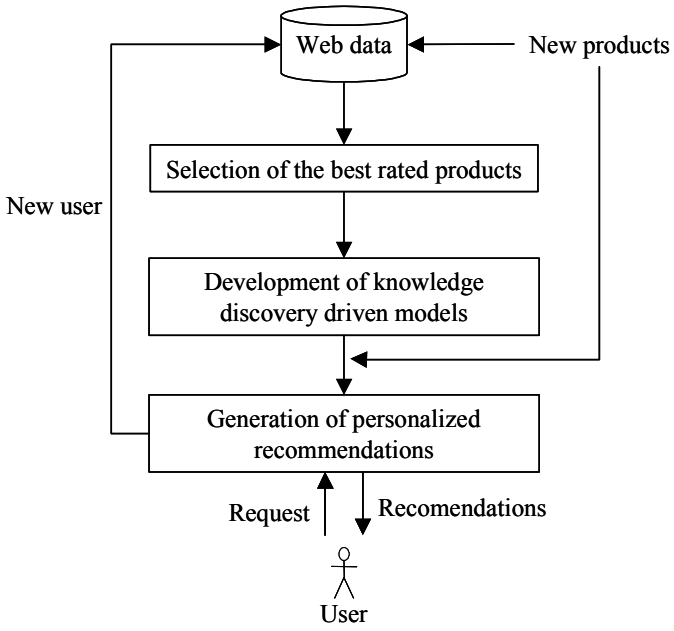

Figure 1. Simplified architecture of the recommender system

4 Experimental Data Description

The experimental study was carried out using data from MovieLens recommender system. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The database contains user rating information about movies, collected through the MovieLens web site (movielens.umn.edu) during a seven-month period. All movies belong to 18 different genres. User ratings were recorded on a numeric five point scale. Users who had less than 20 ratings or did not have complete demographic information were removed from the data set. It consists of:

- 100,000 ratings (1-5) from 943 users on 1682 movies.
- Each user has rated at least 20 movies.
- Simple demographic info for the users (age, gender, occupation, zip)

The files used in this study were:

u.data: The full u data set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of:

user id | item id | rating | timestamp. The time stamps are UNIX seconds since 1/1/1970 UTC .

u.item: Information about the items (movies); this is a tab separated list of:

movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure |

Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |

The last 19 fields are the genres, a 1 indicates the movie is of that genre, a 0 indicates it is not; movies can be in several genres at once. The movie ids are the ones used in the u.data data set.

The statistical study of some of these attributes is shown in figure 2. The projects are between 726 and 8,888 LOC. There are more records with low values of the attributes. However, high values have sufficient weight to influence the induction of the models.

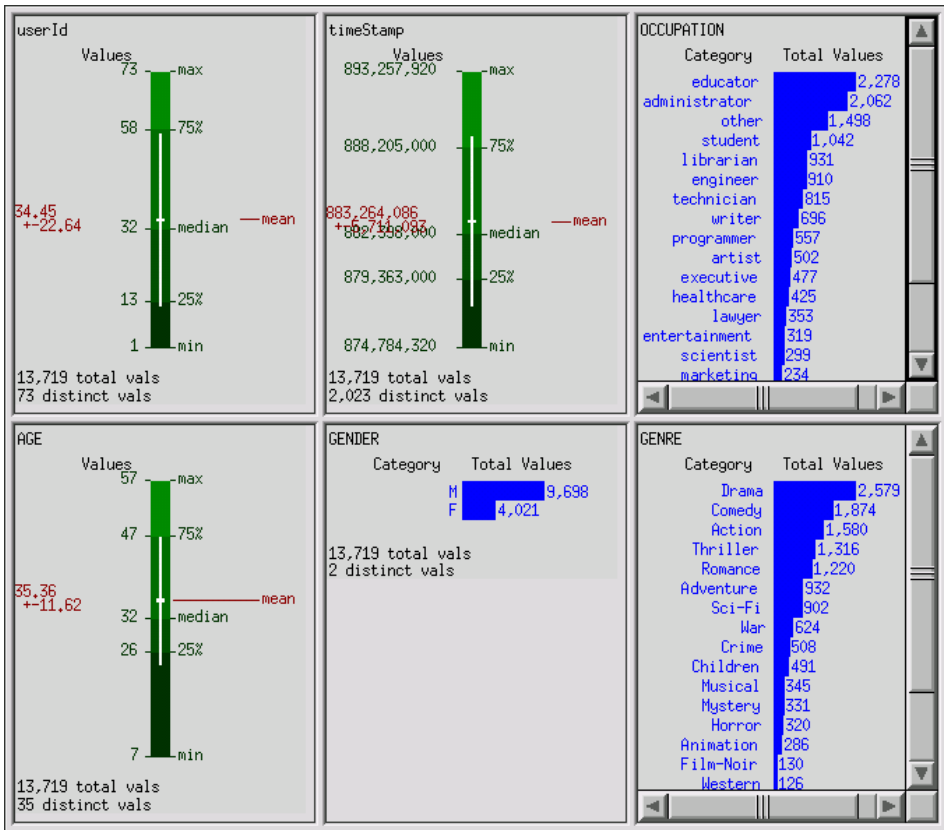

Figure 2. Statistical distribution of the experimental data values

5 Association analysis

5.1. Decision rules and unexpectedness

Methods for rule discovery are widely used in many domains. They have been adopted to target marketing or personalized recommendation service within the e-commerce area.

Now, we introduce the basis of decision rules and the concepts of unexpectedness [12]. A set of discrete attributes $I=\{i_1, i_2, \ldots, i_m\}$ is considered. Let $D=\{T_1, T_2, \ldots, T_N\}$ be a relation consisting on N transactions $T_1, \ldots, T_N$ over the relation schema $\{i_1, i_2, \ldots, i_m\}$. Also, let an atomic condition be a proposition of the form $value_1 \leq attribute \leq value_2$ for ordered attributes and $attribute = value$ for unordered attributes where $value$, $value_1$ and $value_2$ belong to the set of distinct values taken by attribute in D. Finally, an itemset is a conjunction of atomic conditions. In [12] rules and beliefs are defined as extended association rules of the form $X \rightarrow A$, where X is the conjunction of atomic conditions (an itemset) and A is an atomic condition. The strength of the association rule is quantified by the following factors:

Confidence or *predictability*. A rule has confidence *c* if *c*% of the transactions in D that contain X also contain A. A rule is said to hold on a dataset D if the confidence of the rule is greater than a user-specified threshold value chosen to be any value greater than 0.5.

Support or *prevalence*. The rule has support *s* in D if *s*% of the transactions in D contain both X and A.

Expected predictability. This is the frequency of occurrence of the item A. So the difference between expected predictability and predictability (confidence) is a measure of the change in predictive power due to the presence of X [2].

In [11] unexpectedness is defined by starting with a set of rules that represent knowledge about the domain. A rule A → B is defined to be unexpected with respect to the rule X → Y on the database D if the following conditions hold:

- B and Y logically contradict each other (B AND Y |= FALSE);
- A AND X holds on a ''large'' subset of tuples in D;
- The rule A, X → B holds.

For example, an initial rule X → Y is that women like comedy movies (women → comedy). The rule scientist → documentary is unexpected with respect to the initial rule if:

- Comedy AND documentary |= FALSE
- Women and scientist holds on a large subset of tuples on the database
- The rule women, scientist → documentary holds.

Given a belief and a set of unexpected patterns, Padmanabhan and Tuzhilin refine the belief using the discovered unexpected patterns. In the same paper they demonstrate formally that the refined rules have more confidence than the original ones.

5.2. Importance of columns

In classification problems, the *label* attribute is the target of the prediction process. By constructing a relation model between the label and the other attributes, the model can make predictions about new, unlabeled data. Importance of columns is a technique that determines how important various attributes (columns) are in discriminating the different values of the label attribute.

A measure called *purity* (a number from 0 to 100) informs about how well the columns discriminate the classes (different values of the label attribute). It is based on the amount of information (entropy) that the column (attribute) provides. The expression for that information (I) is:

$$\mathrm{I}\,(P(c_1), \ldots, P(c_n)) = \sum_{i=1}^{n} - P(c_i) \log_n P(c_i)$$

where $P(c_i)$ is the probability of the class i and n is the number of classes

If the probabilities of the classes are the same then the information is 1.

The purity is defined as:

$$\text{Purity} = 1 - \mathrm{I}$$

The cumulative purity is a measure of the purity of partitioning the data when more than one column is used. The data are partitioned using columns which influence the classification, in other words, columns which lead to a high purity partition. Each set in the partition has its own purity measure, and the purity of the partition is a combination of these individual measures. For a given set in the partition, the purity is 0 if each class has equal weight, and 100 if every record belongs to the same class. Similarly, the cumulative purity will be 0 if each set in the partition has an equal representation of classes, and 100 if each set in the partition contains record that all have the same class [7].

In our case, we use this method to find the best attributes for discriminating the GENRE label. We searched for the four best attributes. The attributes found by means of the *Mineset* tool [7] were OCCUPATION, AGE, and GENDER. These attributes were used in the refinement of the association rules.

5.3. Refinement process

The refinement of association rules provides recommender systems with more confident rules and serves to solve conflicts between rules. In [11] the beliefs can either be obtained from the decision maker or induced from the data using machine learning methods. In our case, those beliefs were generated from the entire data-base by an association rule algorithm. In an earlier work [9] we have used several visualization techniques and data mining methods to build and validate models for software size prediction. We found that the best attributes for classification give good results in the refinement of associations rules in the

area of projects management. In this paper we propose a recommender procedure that follows the approach of using these attributes in a rules' refinement algorithm wich works with web usage data.

We start with a set of beliefs which relate items. In our case items are genres of movies and users' attributes. The initial beliefs were generated applying an association rules technique to the available data. Then we search for unexpected patterns that could help us to increase the confidence or to solve ambiguities or inconsistencies between the rules representing the beliefs.

The refinement process fits into a generic iterative strategy [12]. Each iteration consists of three steps:

1. Pattern generation procedure: generation of unexpected patterns for a belief.
2. Selection procedure: selection of a subset of unexpected patterns that will be used to refine the belief.
3. Refinement procedure: refining the belief using selected patterns.

The process ends when no more unexpected patterns can be generated. The beliefs are checked at the end of each iteration in order to know if they have acceptable support and confidence.

This generic refinement strategy can be viewed as a broad framework that can allow for a large number of different refinement approaches. We have instantiated this generic strategy and created a specific refinement process [10]. The steps to be taken are described below:

1. Obtain the best attributes for classification and create the sequence: seqA = < A_k >, $k = 1 \ldots t$ (t: number of attributes). The attributes in the sequence are ordered from greater to lesser purity.
2. The values of the attribute A_k are represented as $\{V_{k,l}\}$, $l = 1 \ldots m$ (m: number of different values).
3. Set $k = 1$ and establish the minimal *confidence* c_{min} and minimal *support* s_{min}.
4. Generate initial beliefs with confidence $c \geq c_{min}$ and support $s \geq s_{min}$.
5. Select beliefs with *confidence* near c_{min} or with conflicts between each other:
   Let $X_i \rightarrow Y_i$ and $X_j \rightarrow Y_j$ be two beliefs, R_i and R_j respectively. There is a conflict between R_i and R_j if $X_i = X_j$ and $Y_i \neg = Y_j$.
6. With the selected beliefs create the rule set setR = $\{R_i\}$, $i = 1 \ldots n$ (n: number of selected beliefs)
7. For all beliefs $R_i \in$ setR do:
   7.1. Use the values $\{V_{k,l}\}$ of the attribute A_k for generating unexpected pattern fulfilling conditions of unexpectedness and *confidence* $\geq c_{min}$. The form of the patterns is: $V_{k,l} \rightarrow B$.
   7.2. Refine the beliefs by searching for rules R' like:
   $X_i , V_{k,l} \rightarrow B$
   $X_i , \neg V_{k,l} \rightarrow Y_i$
   7.3. Let setR' be the set of refined rules, then the beliefs refined in step 7.2 should be added to it:
   setR' = setR' $\cup$ { R'_u}, $u = 1 \ldots f$ (f: number of refined rules obtained in the iteration i).
8. Set $k = k + 1$ and setR = setR'.
9. Repeat steps 7 and 8 until no more unexpected patterns can be found.

The principal feature of our approach is the gradual generation of the unexpected patterns by taking a single attribute in each iteration. We take advantage of knowledge of good attributes for classification (see section 5) and use them progressively, beginning with the best. This simplifies the selection of patterns and the refinement process.

6. Experimental data treatment

Initially, the rating information was used to extract the better valued movies. Association rules were produced by taking the records with a rate value greater than 2.

Rules representing the beliefs were generated and visualized by using *Mineset*, a Silicon Graphics tool [7]. The initial beliefs are used to seed the search for unexpected patterns. Figures 3 and 4 are graphical representations of first and refined rules respectively on a grid landscape with left-hand side (LHS) items on one axis (genre-time stamp), and right-hand side (RHS) items on the other (user attributes). Attributes of a rule (LHS $\rightarrow$ RHS) are displayed at the junction of its LHS and RHS item. The display includes bars, disk and colors whose meaning is given in the graph.

Initial beliefs represented in the figure 3 are rules that relate "Genre-Time Stamp" with the user's occupation. Time stamp is the time that the user spent in a product. It is another way of rating the products. Refined rules (figure 4) use the attribute "age" combined with "occupation" as the right-hand side item (RHS).

Rules generator does not report rules in which the predictability (confidence) is less than the expected predictability, that is, the result of dividing predictability by expected predictability (pred_div_expect) should be greater than one. Good rules are those with high values of pred_div_expect

(predictability/expected predictability).The graphs show that these values increase in the refined rules. We have also specified a minimum predictability threshold of 30%.

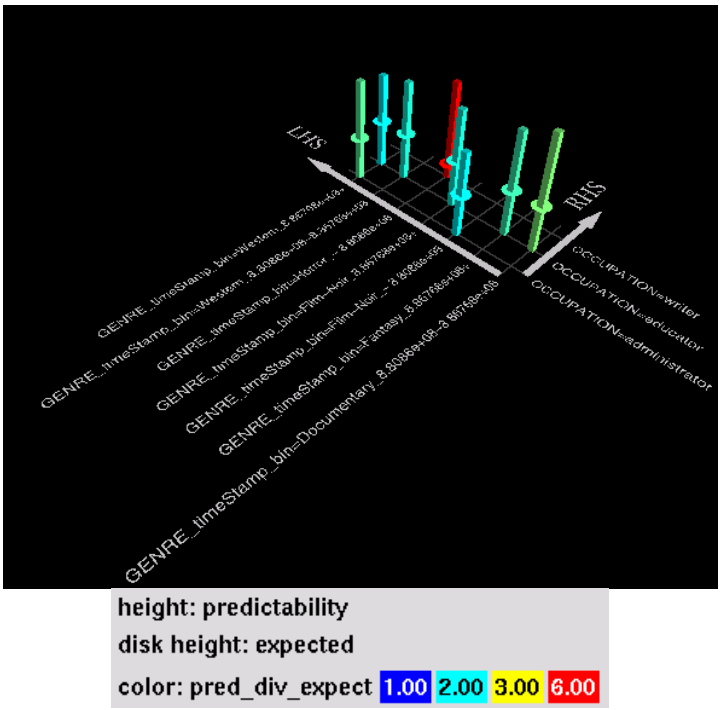

figure 3. Initial beliefs

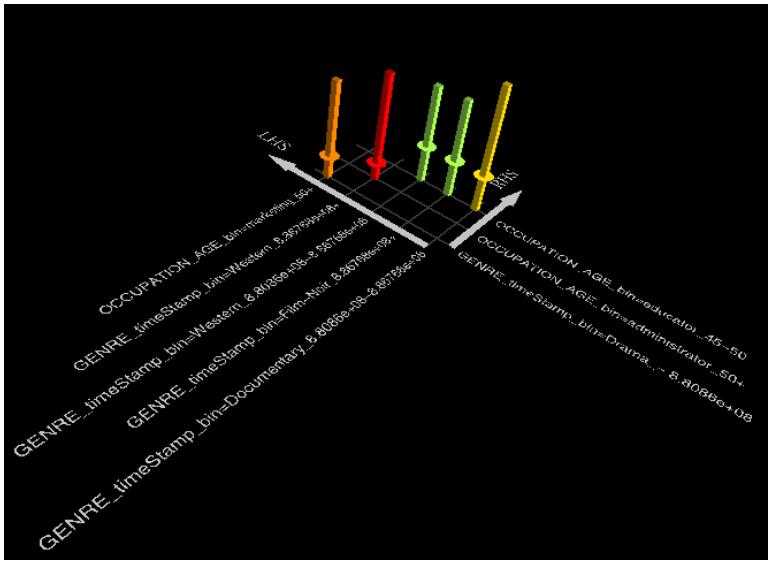

figure 4. Refined beliefs

8 Conclusions

The personalized recommendation engines supply an additional value to the e-commerce systems. They contribute to improve the service to the clients and the competitiveness. E-commerce applications that incorporate recommender systems provide users with intelligent mechanisms for selective retrieval of web information. In this paper, a methodology for recommendation is proposed. This proposal should provide systems with more relevant behaviour patterns which lead to more effective recommendations. The methodology deals with the case of making recommendations for new users. When a new customer accedes to the system, he is checked to obtain his profile and to generate recommendations for him

The goal is to generate strong association rules between attributes that can be obtained from a database that contains web usage information. To do that we have instantiate a rule refinement framework based on discovering unexpected patterns for a belief. By this means we have created a specific process that is very appropriate for the recommender systems area. The identification of weak rules representing beliefs and conflicts between them is the starting point to the iterative refinement process.

The generation of the unexpected patterns is gradual, by taking a single attribute in each iteration. We use information about good attributes for classification and take them progressively, beginning with the best. This simplifies the selection of patterns and the refinement process because the number of patterns generated in each iteration is less.

Rules obtained from the refinement procedure have higher values of confidence and predictability divided by expected predictability. This means that the recommendations are more accurate and the number of false positive recommendations is reduced. Obviously, the correct predictions are not the 100%, but the errors are significantly reduced.

The refinement method has also been successfully applied in the early software size estimation [10]. In this area we have an additional problem: it is necessary to discretize the continuous attributes by splitting the range of values into a manageable number of intervals in order to generate the rules.

The validity of the method has been checked by using the experimental data from the available database. Our next purpose is to prove it in a real e-commerce system and to inquire into the satisfaction of the users.

References

1. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J. and Zanasi, A.:. *Discovering data mining. from concept to implementation*, Prentice Hall, (1998).
2. Cheung, K.W., Kwok, J.T., Law, M.H. and Tsui, K.C.: Mining customer product ratings for personalized marketing. Decision Support Systems, 35 (2003) 231-243.
3. Cho, H.C., Kim, J.K., Kim, S.H.: A Personalized Recommender System Based on Web Usage Mining and Decision Tree Induction. Expert Systems with Applications 23 (2002), 329-342.
4. Hill, W., Stead, L., Rosenstein, M. and Furnas, G.: Recommending and Evaluating Choices in a Virtual Community of Use. Proc. of the Conference on Human Factors in computing Systems-CHI'95, (1995).
5. Konstant, J. Miller, B., Maltz, D., Herlocker, J. Gordon, L. and Riedl, J.: GroupLens: Applying Collaborative Filtering to usenet news. Communications of the ACM, 40 (1997), 77-87,.
6. Lee, CH., Kim, Y.H., Rhee, P.K.: Web Personalization Expert with Combining collaborative Filtering and association Rule Mining Technique. Expert Systems with Applications 21 (2001), 131-137.
7. Mineset user's guide, v. 007-3214-004, 5/98. Silicon Graphics (1998).
8. Mobasher, B., Cooley, R. and Srivastava, J.: Automatic personalization based on web usage mining, Communications of the ACM, 43 (8) (2000), 142-151.
9. Moreno, M.N., Miguel, L.A., García, F.J., Polo, M.J.: Data Mining Approaches for Early Software Size Estimation. Proc. 3rd ACIS International Conference On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'02), 361-368, Madrid, Spain (2002).
10. Moreno, M.N., Miguel, L.A., García, F.J., Polo, M.J.: Building Knowledge Discovery-Driven Models for Decision Support in Project Management. Decision Support Systems, (in press).
11. Padmanabhan, B., Tuzhilin, A.: Knowledge Refinement based on the discovery of unexpected patterns in datamining. Decision Support Systems 27 (1999) 303– 318.
12. Padmanabhan, B., Tuzhilin, A.: Unexpectedness as a measure of interestingness in knowledge discovery. Decision Support Systems 33 (2002) 309– 321.
13. Resnick, P., Iacovou, N., Suchack, M., Bergstrom, P. and Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. Proc. Of ACM CSW'94 Conference on Computer.Supported Cooperative Work, 175-186, (1994).
14. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithm. Proceedings of the tenth International World Wide Web Conference (2001), 285-295.
15. Schafer, J.B., Konstant, J.A. and Riedl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery, 5 (2001), 115-153.
16. Shardanand, U. and Maes, P. "Social Information Filtering: Algorithms for automating 'Word of Mouth'. Proc. of the Conference on Human Factors in computing Systems-CHI'95, 1995
17. Weiss, S.M., Indurkhya, N.: *Predictive data mining. A Practical Guide*, Morgan Kaufmann Publishers, San Francisco, (1998).
18. Wolf, J., Aggarwal, C. Wu, K.L. and Yu, P.: Horting Hatches an Egg. A new graph-theoretic Approach to Collaborative Filtering. Proc. Of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, C.A., (1999).