A Bayesian Network approach to Hybrid Recommending Systems

Luis M. de Campos **Juam M. Fernández-Luna** **Juan F. Huete**
Dpto. Ciencias de la Computación e I.A.
Universidad de Granada
18071 – Granada, Spain.
{lci,jmfluna,jhg}@decsai.ugr.es

Abstract

The problem of building Recommender Systems has attracted considerable attention in recent years. The objective of this paper is to automatically recommend and rank a list of new items to a user based on both items this user has already voted on and past voting patterns of other users. The proposed model is based on Probabilistic Computing, one of the multiple fields that comprise Soft Computing. Particularly we make use of the Bayesian networks formalism.

Keywords: Recommender System, Bayesian Networks.

1 Introduction

The change of the web use, from an academic use to a commercial one, has been carried out during the last decade. This commercialization of the web presents a new set of problems to be solved and therefore new opportunities for researching. In this paper we will study the following problem: How to design a system that can help to recommend new products to clients?.

This kind of systems are called *Recommender Systems* (RS). Broadly speaking, a RS provides specific suggestions about items (or actions), within a given domain, which may be considered interesting to the user [13]. In order to recommend new products, the system takes into account the user preferences as well as the similarities of the user with other users in the system (obtained from past voting patterns).

The set of user's preferences, known as the *user profile*, is normally provided either explicitly (by means of a form or a questionnaire when he/she logs in) or implicitly (using purchase records, viewing or rating items, visiting links, taking into account the membership to a certain group,...). These preferences will be collected over time, giving a much more reliable identification of the user preferences or tastes.

The proposed system will be modeled using Bayesian networks (BN). The reason is that we can combine a qualitative representation of the problem (which explicitly represents the dependence and independence relationships between items and user votes, as well as the dependences or similarities between users) with a quantitative representation by means of a set of probability distributions, measuring the strength of these relationships. In all the process we must consider the computational aspects of the RS, where the sparseness of the data and the fact that the ranking should be computed in real time constitute two challenging problems.

The paper is organized in the following way: Section 2 presents the fundamentals of RS as well as related works. Section 3 describes the format of the input databases. Section 4 is used to describe the topology of the BN and its learning algorithm. Section 5 discuss the estimation of the confitional probability distributions from the datasets. Section 6 shows how the inference is performed in order to give recommendations to the user. Section 7 presents some experimental results and Section 8 includes the conclusions and some remarks about further research.

2 Recommender Systems

Although many other approaches to the RS are found in the literature [9], in this paper we focus on probabilistic model-based approaches to RS. The predictions are made by building (offline) an explicit model of the relationships among items. This model is then used (online) to finally recommend the product to the users. There are three main variants of Recommender Systems [1]:

• *Content-based RS*: In this variation the system stores content information about each item to be recommended. This information will be used to estimate how similar certain items are to each other or the similarity with respect to user preferences (also represented by means of a subset of content features).

Different approaches can be found in the literature. For instance, in [11] a probabilistic user model is constructed based on the content description of the items rated and then is used to infer ratings for unseen items. Search engines can be also be thought as recommender systems purely based on content description [4]. Based on BN, [6, 7] recommend a product by taking two different (but complementary) situations into account: the exhaustivity and the specificity of the product with respect to the user needs. In addition, influence diagrams [5] have been used to deal with RS in hierarchical domains where the items to be recommended can be grouped in a hierarchy.

• *Collaborative filtering RSs*: These systems try to identify groups of people with tastes similar to that of the user and recommend items that they have liked. In this case, we can distinguish between two approaches. The first one, that construct a full joint probability distribution about items and then uses this distribution to make prediction online. For instance, in [15, 3] BN learning algorithms are the tools by which the distribution is learned. Also BN-based classifiers [2, 10, 14] have been applied to this purpose. The second approach builds several conditional models, one for each user, that predict the likelihood of an individual item given a combination of the observed votes for the rest of users [8].

• *Hybrid RS*: In this case, both content and collaborative approaches are combined. An example of a hybrid collaborative filtering is proposed for help on browsing documents on line [12]. In this model a joint density function is constructed assuming the existence of a hidden variable representing the different topics of the documents. This variable renders users, documents and words conditionally independent. The probability parameters relating the hidden variable with the observed data are estimated using EM algorithm.

Hybrid systems have the advantage of both, content-based systems because they can make recommendations for entirely new items for which there is no history, and collaborative systems since they do not ignore historical potential valuable information.

3 The database

The systems that we propose in this paper lyes in the third approach of RS, so we have to consider two different information sources: the content-based knowledge and the collaborative component.

With respect to the content component we typically have a large number m of items $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ and a large number l of features $\mathcal{F} = \{F_1, F_2, \ldots, F_l\}$, where each item has been described by a set of attributes or features. For instance, in order to describe a movie we can use a set of genres (action, western, comedy, etc.) or to describe a restaurant the kind of food it serves (Mediterranean, Chinese, etc.) can be used. In this case, the content description can be viewed as a very sparse binary $m \times l$ matrix $\mathbf{D}$, where $d_{i,j} = 1$ indicates that item i can be described with feature j, and the entry is assumed to be zero otherwise. For instance, consider the following matrix description where rows correspond to items and columns correspond to features.

Table 1. Database of Items description, **D**

$\mathcal{I}$	0	1	2	3	4	5	6	7	8	9
I_1	0	1	1	0	1	0	0	0	0	0
I_2	0	0	0	1	1	0	0	0	0	0
I_3	0	0	1	1	0	1	0	0	0	0
I_4	0	0	0	1	1	0	0	0	0	0
I_5	0	0	0	0	0	1	0	0	1	0
...	.	.	.	.	.	.	.	.	.	.

With respect to the collaborative component, we have a large set of n users, $\mathcal{U} = \{U_1, U_2, \dots, U_n\}$, which have rated some items, either explicitly or implicitly. We denote by $\#r$ the number of different values that can be used to rate an item (e.g., a movie can be rated on a scale of 1 to 5). Again, the observed data can be viewed as a very sparse $n \times m$ matrix, **R**, since a typical user only rates a very small fraction of the items. In the matrix **R**, $r_{a,j}$ represents the rate of user U_a for the item I_j and it is assumed to be zero when the item is no voted on by the user. For example, consider the following matrix where the rows represent the users, columns are used to represent items and $\#r = 2$.

Table 2. Database of user rates, **R**

$\mathcal{U}$	I_1	I_2	I_3	I_4	I_5	...
U_1	2	2	0	1 0	0	.
U_2	0	0	1	2	0	.
U_3	2	2	0	0	0	.
U_4	2	1	0	0	0	.
U_5	0	0	0	0	2	.
...	.	.	.	.	.	.

4 BN hybrid RS: Setting the dependence relations

To design the system we adopt the following set of 'linear' relations between information sources:

$$\mathcal{F} \longrightarrow \mathcal{I} \longrightarrow \mathcal{U} \longrightarrow \mathcal{V}$$

where the relation $\mathcal{F} \longrightarrow \mathcal{I}$ is used to model the description of the items $\mathcal{I}$ by means of a set of features $\mathcal{F}$, representing the content based component of the model. The relation $\mathcal{I} \longrightarrow \mathcal{U}$ models the database of user votes or ratings for the observed items. Finally, the relation $\mathcal{U} \longrightarrow \mathcal{V}$, which represents the collaborative component, will be used to represent that the final prediction for an active user will also depend on the votes of people with similar preferences.

4.1 Content Based Component

This component contains the set of m features, $\mathcal{F}$, and the set of l items, $\mathcal{I}$. There exists a feature node F_k for each feature used to describe a product and, also, there exists a node for each product in the system, I_j.

Both, feature F_k and item I_j, have associated a random binary variable, taking values from the sets $\{f_{k,0}, f_{k,1}\}$ and $\{i_{j,0}, i_{j,1}\}$, respectively. The subindex 0 is used to represent that the variable is not-relevant to the user's preferences whereas the subindex 1 represents the relevance alternative.

With respect to the topology of this (sub)model, we consider the following logical implication: Since a product is described with a fixed set of features, there is an arc from each feature node to each item node having this features (Figure 1 shows the relations that can be obtained from Table 1). With these arcs we are expressing the fact that the probability of relevance of the product will depend on the relevance values of the different features that comprise it[1].

4.2 Modeling the User Votes

In order to represent the dependence relationships between items, $\mathcal{I}$, and user votes, $\mathcal{U}$, we have to include one variable for each user which had voted for a product and a set of arcs going from each product, I_j, which has been voted by the user U_a to the node representing that user (Figure 1 shows the relations expressed in Table 2). We do not include arcs connecting features and user votes because we consider that these variables are independent

[1]Although the presented topology implies that a feature F is marginally independent of any other feature, this assumption (restrictive in some domains) could be relaxed to include relationships between evidence items.

given that we know the relevance value for the items in the system.

A node, U_a, will be used to represent the probability distribution associated to the pattern of rating of user U_a. If we assume that votes are integer scaled with a range from 1 to $\#r$, the states of each node correspond with the set $\{0, 1, 2, \ldots, \#r\}$. Note that there exits an additional value 0 which has been included to model the "No interest on vote" situation.

4.3 Collaborative Component

This component is used to relate a given user with other users with similar tastes. These relationships should be directly modeled in the BN by means of the inclusion of arcs between any two related users. These arcs would be obtained by means of a learning algorithm that searches for dependences between users. Thus, whenever a dependence (similarity) between the preferences of user U_a and user U_b has been found, an arc connecting both nodes will be included in the BN. But, taking into account that similarities between users tastes tends to be symmetric (when U_a is highly related with U_b it is common for U_b being also related with U_a), a cycle could be included in the BN, which is forbidden in a BN topology.

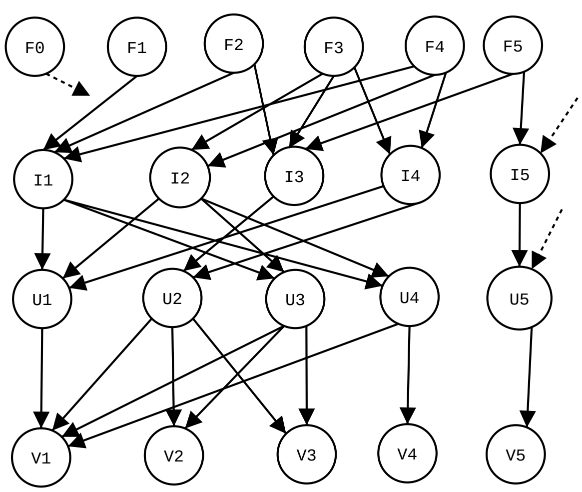

Figure 1: Recommending System Topology

So, to facilitate the presence of these relationships in the model, we propose the inclusion of a new of set of nodes $\mathcal{V}$ to denote collaborative votes. There exists one collaborative node for each user in the system, i.e., $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$. These nodes will be also used to estimate the probability distributions of the user's votes and therefore they will take their values in the same domain than $\mathcal{U}$, i.e., $\{0, 1, 2, \ldots, \#r\}$.

The parent set of a variable V_a in the graph, $Pa(V_a)$, will be learned from the database of votes, $\mathbf{R}$. This set will contain those user variables, $U_b \in \mathcal{U}$, where U_a and U_b should have a greater similarity between their tastes. Naturally, U_a should belong to $Pa(V_a)$. Thus, given a similarity measure, the set $Pa(V_a)$ can be obtained by using a threshold or considering only the first p variables in the ranking. We propose to use the Pearson correlation coefficient to measure the similarity between users, also used as the basis of the weights in different collaborative systems.

$$sim(U_a, U_b) = \frac{\sum_j (r_{a,j} - \overline{r}_a)(r_{b,j} - \overline{r}_b)}{\sqrt{\sum_j (r_{a,j} - \overline{r}_a)^2 \sum_j (r_{b,j} - \overline{r}_b)^2}} \quad (1)$$

where the summations over j are over those items for which users U_a and U_b have recorded votes, $Pa(U_a) \cap Pa(U_b)$ in the BN. If there are no common items in U_a and U_b voting histories, then $sim(U_a, U_b) = 0$ by default. Also, $\overline{r}_a$ is the mean vote for user U_a, i.e.,

$$\overline{r}_a = \frac{1}{|Pa(U_a)|} \sum_{I_k \in Pa(U_a)} r_{a,k}.$$

5 Estimating the conditional probability distributions

To complete the model's specification, the numerical values for the conditional probabilities have to be assessed. Previously, we are going to introduce some notation. Thus, given a variable X_i, lowercase letters are user to denote the variables realization. For instance, $x_{i,j}$ denotes that variable X_i takes the j^{th}-value. We write $Pr(x_{i,j}|pa(X_i))$ for $P(X_i = x_{i,j}|pa(X_i))$, with $pa(X_i)$ denoting a configuration of the parent set of X_i, $Pa(X_i)$, or sometimes $Pr(X)$ to denote the probability distribution.

Whereas the estimation of the probability distribution for feature nodes is quite simple, the situation for the rest of the probability distributions is more complex because the great number of parents a node might have. In these cases, it can be quite difficult to assess and store the required conditional probability tables (with a size exponential with the number of parents). So, we propose the use of a canonical model, similar to the one presented in [4], to represent the conditional probabilities, which will allow us to design a very efficient inference procedure. Thus, for a given node X_i we have to define these probabilities as follows:

$$Pr(x_{i,j}|pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \quad (2)$$

where $y_{k,l}$ is the value that variable Y_k takes in the configuration $pa(X_i)$ and $w(y_{k,l}, x_{i,j})$ are weights measuring how this l^{th} value of variable Y_k describes the j^{th} state of node X_i, with $w(y_{k,\bullet}, x_{i,j}) \geq 0$ and $\sum_{Y_k \in Pa(X_i)} w(y_{k,\bullet}, x_{i,j}) \leq 1$.

All these weights might be estimated from the datasets and could be defined as follows:

– For every feature F_k , since it has no parent, we need to assess the a priori probability of relevance. In this paper we propose $Pr(f_{k,1}) = n_k/m$, being m is the number of products (the size of $\mathcal{I}$) in the dataset and n_k is the number of times that feature F_k has been used to describe a product. The value $Pr(f_{k,0})$ is obtained as $Pr(f_{k,0}) = 1 - Pr(f_{k,1})$.

For instance, considering the data in Table 1, we have that $Pr(f_{2,1}) = 0.2$ and that $Pr(f_{5,1} = 0.4)$

- Given an item node, I_j in $\mathcal{I}$, since it represents a bivaluated variable, we only have to define the weights needed to compute $Pr(i_{j,1}|pa(I_j))$ because $Pr(i_{j,0}|pa(I_j)) = 1 - Pr(i_{j,1}|pa(I_j))$.

$$\begin{array}{l} w(f_{k,1}, i_{j,1}) = \frac{log((m/n_k)+1)}{log(m+1)M(I_j)} \\ w(f_{k,0}, i_{j,1}) = 0 \end{array} \quad (3)$$

being $M(I_j)$ a normalizing factor computed by means of $M(I_j) = \sum_{F_k \in Pa(I_j)} log((m/n_k) + 1)/log(m + 1)$. The expression $log((m/n_k) + 1)/log(m + 1)$ is used to measure the importance of the feature in the whole data set, in a similar way than *inverted document frequency* is used in the field of Information Retrieval. Note that the more features are relevant in the configuration the greater the probability of relevance of I_j.

For example, using the data in Table 1, $M(I_1) = 2.247$, $w(f_{1,1}, i_{1,1}) = 0.445$, $w(f_{2,0}, i_{1,1}) = 0$ and $w(f_{4,1}, i_{1,1}) = 0.243$. Then, $Pr(i_{1,1}|f_{1,1}, f_{2,0}, f_{4,1}) = 0.706$.

- For user's node $\mathcal{U}$, we need to measure the influence of the item I_k in the pattern of vote of U_a. Thus, if user U_a votes for item I_k with the value s, i.e., $r_{a,k} = s$, and depending whether the configuration $pa(U_a)$ represents that the user has interest in the item I_k or not ($I_k = i_{k,1}$ or $I_k = i_{k,0}$, respectively), these weights will be defined as.

$$\begin{array}{l} w(i_{k,1}, u_{a,s}) = 1/|Pa(U_a)|. \\ w(i_{k,1}, u_{a,t}) = 0, \text{with } t \neq s, 0 \leq t \leq \#r. \\ w(i_{k,0}, u_{a,0}) = 1/|Pa(U_a)|. \\ w(i_{k,0}, u_{a,t}) = 0, \ 1 \leq t \leq \#r. \end{array} \quad (4)$$

being $|Pa(U_a)|$) the number of items previously voted by U_a.

For instance, given $pa(U_1) = \{i_{1,1}, i_{2,0}, i_{4,1}\}$ and the data in Table 2, we have that the values of the distribution $Pr(u_{1,k}|pa(U_1)$ are: $Pr(u_{1,0}|pa(U_1)) = 0 + 1/3 + 0$, $Pr(u_{1,1}|pa(U_1)) = 0 + 0 + 1/3$ and $Pr(u_{1,2}|pa(U_1)) = 1/3 + 0 + 0$.

- Focusing on collaborative vote nodes $\mathcal{V}$, to estimate the final vote for the user V_a we have to distinguish between the weight that user U_a brings by himself and those weights given by the users with similar tastes U_b, with $U_b \in pa(V_a)$.

$$\begin{array}{l} w(u_{a,s}, v_{a,s}) = \alpha \\ w(u_{a,t}, v_{a,s}) = 0, \text{with } t \neq s, 0 \leq t \leq \#r. \end{array} \quad (5)$$

and for $0 \leq t, s \leq \#r$

$$w(u_{b,t}, v_{a,s}) = \frac{(1-\alpha)}{|Pa(V_a)|-1} \frac{N(u_{b,t}, v_{a,s}) + \beta q_s}{N(u_{b,t}) + \beta}. \quad (6)$$

where $0 \leq \alpha \leq 1$ in such a way that the greater is α the greater weight is assessed to the past votes of U_a. The value $N(u_{b,t}, v_{a,s})$ is the number of items from the set $Pa(U_a) \cup Pa(U_b)$ that being voted with value t by user U_b have been also voted with value s by user U_a and $N(u_{b,t})$ is the number of items in $Pa(U_a) \cup Pa(U_b)$ voted with value t by user U_b. β and q_s are the parameters of a Dirichlet prior over user ratings. Note that the weights $w(u_{b,t}, v_{a,s})$ are proportional to the maximum a posteriori (MAP) estimator of $Pr(v_{a,s}|u_{b,t})$.

For example, considering $\alpha = 0.5, \beta = 1$ and $q_i = 1/3$ and the data set in Table 2, we have that for the configuration $pa(V_1) = \{u_{1,2}, u_{2,0}, u_{3,1}, u_{4,1}\}$ the probability distribution for V_1 is: $Pr(v_{1,0}|pa(V_1)) = Pr(v_{1,1}|pa(V_1)) = 0+0.02+0.06+0.03 = 0.11$ and $Pr(v_{1,2}|pa(V_1)) = 0.5 + 0.12 + 0.05 + 0.11 = 0.78$.

6 Helping Users: Inference

Previously to discuss the inference mechanism we have to discuss how the users should interact with the system . In order to illustrate the process, suppose a RS where the items are a set of movies and the features are the set of genres or keywords describing a movie. In this case, we have the following alternatives:

1. Given an active user U_a and a movie I_k in the system (rated by others users) the problem is to predict the rate that user U_a should assess to the movie. From a probabilistic point of view we have to compute $Pr(v_{a,s}|I_k)$, $1 \leq s \leq \#r$. Note that this is the typical performance of collaborative RS. With the same philosophy, the system can be used to recommend to user U_a those movies with higher probabilities of being liked by him. This can be done by instantiating in turn the set of unseen movies of U_a.

2. Given a new movie which do not belong to the system (it has not been voted previously by any user), we can also predict the rate that a given user U_a should assess. Here, if we consider as evidence those features describing the movie $(F_i, \ldots, F_m)$ and we instantiate them in the BN, we can compute $Pr(v_{a,s}|F_i, \ldots, F_m)$, $1 \leq s \leq \#r$. Also, by computing these probabilities for all users in the system, we can recommend this new movie, as in a information filtering process, to those users who should rate positively the movie.

3. Focusing in the content based component, the user can express also his/her preferences on some features $F_g, \ldots, F_k$ (e.g., 'western and comedy') and by instantiating in turn the unseen movies I_j in the system compute $Pr(v_{a,s}|F_g, \ldots, F_k, I_j)$. Then, the higher rated movies will be recommended.

Therefore, regardeless the user's interest is in a new movie or in a set of features, and in order to homogenize the inference process, we always consider as evidence the instantiation of a subset of features, those used to describe either the user's preferences explicitly and/or the movie(s).

With this instantiation we guarantee that the estimation of the final score of a user U_a will receive evidences from two different paths: On the one hand, the one that consider the similarity between the new movie and those movies that has been previously rated by him. On the other hand, by estimating how other users of the system would rate the new movie and considering the similarity of the user U_a with them.

6.1 Inference

In order to predict the utility of unseen items for user U_a we have to compute the probability distribution $Pr(v_{a,s}|ev)$, $0 \leq s \leq \#r$. From a probabilistic perspective, this probability can be defined by means of

$$Pr(v_{a,s}|ev) = \sum_{\mathcal{F},\mathcal{I},\mathcal{U}} Pr(v_{a,s}, F_l, I_k, U_j|ev). \tag{7}$$

Considering (1) that in a BN a node is independent of all its predecessors given that we know the the particular values that its parents takes and (2) using the advantages of the canonical models used to express the conditional probability distributions (equation 2), these final probabilities can be computed efficiently by means of a top-down propagation mechanism, where the posterior probabilities of one layer are obtained using the posterior probabilities computed in the previous layer, starting from features nodes.

The following theorem (the proof is omitted due to lack of space) shows how to compute the a posteriori probabilities.

Theorem 1: *Let X_a be a node in the RS network, let m_{X_a} be the number of parents of X_a, let Y_j be a node in $Pa(X_a)$ and l_{Y_j} the number of states that Y_j takes. Then, if the conditional probability distributions can be expressed under the conditions given by equations 2, the exact a posteriori probabilities can be computed by means of the following formula, where if F_j is not an evidence, $F_j \notin ev$, $Pr(f_{j,1}|ev) = Pr(f_{j,1})$ and $Pr(f_{j,1}|ev) = 1$ otherwise:*

$$Pr(x_{a,s}|ev) = \sum_{j=1}^{m_{X_s}} \sum_{k=1}^{l_{Y_j}} w(y_{j,k}, x_{a,s}) \cdot Pr(y_{j,k}|ev).$$

7 Empirical Analysis.

Althought a more detailed experimentation must be done, this section presents some preliminary results about the performance of the system. To do that we consider the MovieLens dataset collected by the GroupLens Research Project at the University of Minnesota during the seven-month period from September 19th, 1997 through April 22nd, 1998 with votes ranging from 1 to 5[2]. The dataset contains 1682 movies, each one being described by a set of keywords from a set of 19 genres. The number of users in the dataset is 943. We use a set of 20000 rating as the training set to learn both the network topology and the parameters. With respect to the topology we determine that two user U_a and U_b are highly related when the absolute value of the correlation coefficient is greater than 0.85 and to estimate the weights in eq. 6 and 7 we use the values $\alpha = 0.5, \beta = 1$ and $q_i = 1/6$. We also use a set of 4732 rating as the test set. Note that training and test sets are disjoint.

[2]http://www.movielens.org

Particularly, our objective in this experimentation is to study the predictive capacity of the model, thus we use the dependences learned from the training set to predict the votes in the test set. In order to predict the vote for the active user we use two different selection alternatives: In the first one we show the state with higher probability (HP) and in the second one we show the state with better ratio (BR) between the a priori (without evidences) and the a posteriori probabilities.

Also, in order to predict the performance of the system we use two different measures: A(•) which measures the percentage of times that the system predict the correct vote of the user, using both HP and BR selection criteria and AD(•) which measures the average absolute deviation between the recommended and the true user votes.

Table 3. Experimental Results.

A(HP)	A(BR)	AD(HP)	AD(BR)
0.501	0.322	0.815	1.094

From this table we can conclude that recommending the state with higher probability is the best option being its performance quite reasonable since the half of the times it has success on predict the correct vote whereas in the other cases the system output is not so far from it.

8 Conclusions

A general hybrid BN-based model for recommendation systems has been proposed in this paper. Taking into account efficiency considerations, the posterior probabilities for recommending are computed using a very efficient method based on canonical models. Guidelines of how estimate the probability values from a data set and also how the RS interacts with the users have been given. It must be note that the proposed model is quite general, since it can be applied to solve different recommendation tasks being also able to incorporate different sources of evidence (see Section 6).

As future works, we are planning to evaluate the model with real problems, involving real users to determine the quality of the recommendations provided. We are also studying mechanisms to incorporate better specifications of the products into the system and new methods to estimate the weights stored in the nodes of the BN.

Acknowledgements

This work has been supported by the Spanish "Ministerio de Educación y Ciencia", under Project TIN2005-02516.

References

[1] M. Balabanovic and Y. Shoham. 1997. Fab: Content-based, collaborative recomendation. *Communications of the ACM*, 40(3):66–72.

[2] J.S. Breese, D. Heckerman, and C. Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, 43–52.

[3] C.J. Butz. 2002. Exploiting contextual independencies in web search and user profiling. In *Proc. of World Congress on Computational Intelligence*, 1051–1056.

[4] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2003. The BNR model: Foundations and performance of a Bayesian network retrieval model. *International Journal of Approximate Reasoning*, 34:265–285.

[5] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2005. A Decision-Based Approach for Recommending in Hierarchical Domains. *Lecture Notes in Artificial Intelligence, 3571* 123-135.

[6] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2005. e-Bay.Net: Helping Users to Buy in e-commerce Applications. *Proc. EUSFLAT05-LFA Conference*, pp 565-570.

[7] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. 2005. Generalizing e-Bay.NET: An Approach to Recommendation Based on Probabilistic Computing. *1er. Workshop on Web Personalization, Recommeder Systems and Int. User Interface*, pp. 24-33.

[8] D. Heckerman, C. Chickering, D.Meek, C. Rounthwaite and C.Kadie Dependency networks for inference, collaborative filtering and data visualization. In *Jour. Mach. Learning Res. 1*, pp. 49-75.

[9] S. Kangas. 2002. Collaborative filtering and recommendation systems. VTT Information Technology, Research report TTE4-2001-35.

[10] K. Miyahara and J. Pazzani. 2000. Collaborative filtering with the simple Bayesian classifier. In *Proc. of the Pacific Rim Int. Conf. on Artif. Intell.*, 679–689.

[11] R. Mooney and L. Roy 2000 Content-based book recommending using learning for text categorization. In *Proc. 5th ACM Conf. on Digital Libraries* pp.195-204

[12] 2001 A. Popescul, L. Ungar, D. Pennock and S.Lawrence. Probbabilistic model for unified collaborative and content-based recommendtation in sparse-data environments. In *Proc. 17th Conf. on Uncert. in Artif. Intellig,* pp. 437-444.

[13] P. Resnick and H.R. Varian. 1997. Recommender systems. *Communications of the ACM*, 40(3):56–58.

[14] V. Robles, P. Larrañaga, J.M. Peña, O. Marbán, J. Crespo, and M.S. Pérez. 2003. Collaborative filtering using interval estimation naive Bayes. *Lecture Notes in Artificial Intelligence*, 2663:46–53.

[15] S.N. Schiaffino and A. Amandi. 2000. User profiling with case-based reasoning and Bayesian network. *Proc. of the Iberoamerican Conf. of Artificial Intelligence*, 12–21.