Group Recommending: A methodological Approach based on Bayesian Networks

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, Miguel A. Rueda-Morales Departamento de Ciencias de la Computación e Inteligencia Artificial E.T.S.I. Informática, Universidad de Granada, 18071 – Granada, Spain {lci,jmfluna,jhg,mrueda}@decsai.ugr.es

Abstract

The problem of building Recommender Systems has attracted considerable attention in recent years, but most recommender systems are designed for recommending items for individuals. The aim of this paper is to automatically recommend and rank a list of new items to a group of users. The proposed model can be considered as a collaborative Bayesian network-based group recommender system, where the group's rates are computed from past voting patterns of other users with similar tastes. The use of Bayesian networks allows us to obtain an intuitive representation of the mechanisms that govern the relationships between the group members.

1 Introduction

Everyone has at some time or another had to think about questions such as what movie to see, where to go on holiday, where to go for lunch, what book to read, and although the possible answers might seem easy this is not always the case due to the numerous possible answers, i.e. the enormous number of movies to see, books to read, places to visit, restaurants to eat at, etc. *Recommender Systems* (RS) have emerged to address this issue. Broadly speaking, an RS provides specific suggestions about items (or actions) within a given domain and which may be considered of interest to the user [22].

There are many kinds of RS, depending on the information used when recommending. In this paper, we will explore the variant called *collaborative filtering* which attempts to identify groups of people with similar tastes to those of the user and to recommend items that these have liked. Most of these RS are designed for individual use, i.e. there is an active user that (once connected with the system) receives recommendations about certain products.

In this paper, we will study a related problem: Group

Recommending, where the objective is to obtain recommendations for groups of people where their members may be inter-related in different ways. This kind of RS is appropriate for domains where a group of people participates in a single activity such as seeing a movie or going on holiday and also situations where a single person must make a decision about a group of people, for example, buying a company gift or deciding on a restaurant to go to. This is a relatively novel problem (research started focusing on group recommending at the start of the 21st century [19, 15]) and has hardly been researched in literature.

The purpose of this paper is to develop a general methodology based on the Bayesian network (BN) formalism for representing both the interactions between group members and the processes leading to the final choice or recommendation. In particular, our objective can be summarized in studying how to determine how a group of people rates a product which is unknown to the members of the group. We will not discuss subjects such as how the groups are formed, how long they have existed, and the relationships between their members. It should be noted that in this problem it is pointless to discuss strategies about how to obtain consensus between the different group members since they did not know the products before.

We consider BNs appropriate because they combine a qualitative representation of the problem by means of an explicit representation of the dependence relationships between items, users and groups, with a quantitative representation by means of a set of probability distributions, measuring the strength of these relationships. Throughout 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.

Thanks to BNs, the predicted group vote depends on the a posteriori joint probability distribution over all the possible rates that the members of the group should give in contrast to naive models where firstly a collaborative RS decides the vote for each user and then each predicted vote is merged using fixed criteria. The novelty of this approach is the representation of different decisions rules (approaches used by a group to mark the choice that is made) by means of conditional probability distributions.

The second section of this paper presents RS fundamentals and related work. Section 3 describes the BN topology, its learning algorithm and the estimation of the conditional probability distributions from the datasets. Section 4 shows how the groups have been modeled. Section 5 explains how the inference is performed in order to provide recommendations to the group. Section 6 presents some experimental results. Finally, Section 7 includes our conclusions and some comments about further research.

2 Related Work

Although many other approaches to the RS can be found in the literature [11], in this paper we will focus on probabilistic model-based approaches to RS.

There are three main variants of RSs [1]. The first variant is content-based RSs[18, 4, 5] which store content information about each item to be recommended and this information will then 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). Collaborative filtering RSs attempt to identify groups of people with similar tastes to those of the user and to recommend items that they have liked. In this case, we can distinguish between two approaches. The first approach constructs a full joint probability distribution about the items and then uses this distribution to make predictions online. For instance, in [24, 3] BN learning algorithms are the tools by which the distribution is learned. BN-based classifiers [2, 17, 23] have also been applied for 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 remaining users [8]. Finally, hybrid RSs are a combination of both content and collaborative approaches. An example of hybrid collaborative filtering can be found in [21, 6].

While the problem of single recommending has been widely explored in literature, the problem of group recommending is relatively new in our field, although the related problem of *group decision making* (deciding what is best for a group given the opinion of the individuals) has been extensively studied in economics, politics, and mathematics. Group recommending systems could be classified into two main categories, depending on whether or not they use member feedback:

Passive members: the purpose is to provide recommendations about a set of items. Here the group members have a passive role in the sense that they do not

evaluate the proposed recommendations, and this is closer to the classical RS approach. In addition, the dichotomy between content-based and collaborativebased group RSs can be found. Examples of contentbased group recommending are [12], which are designed to generate a sequence of songs for a party taking into account the musical tastes of each party goer, and [15], which selects the music stations to be played at a gym. These systems attempt to maximize overall group satisfaction by considering each member's preferences about genres or artists. Related to collaborative-based group RS is Polylens [19], which is an extension of the MovieLens1 [9] system that recommends movies to groups of users. These use nearest neighbor algorithms to find those individuals which are similar to group tastes and to obtain recommendations which merge the voting preferences of these individuals according to the principle of least misery.

· Active members: the objective is to obtain consensus between group members, and is similar to many decision support system approaches which imply that the users might evaluate system recommendations. For instance, [16] is designed to provide assistance to a group of friends trying to plan a skiing vacation. A recommendation is performed using the incremental critiquing method, whereby a compatibility score for the critiques of the user and those of the remaining group members is computed. [10] also helps a group of users to agree on the desired attributes of a vacation that they are planning to take together. In this system, the userspecifications are aggregated to produce group recommendations. In [13, 14, 25], different strategies were studied for combining individual user models to adapt to groups, such as how humans select a sequence of television items to suit a group of viewers.

3 Group Modeling

In this paper we will focus on collaborative-based group RS. Firstly, we will briefly discuss the information source. We therefore typically have a large number m of items $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ and a large set of n users, $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$. Each user has rated certain some items, either explicitly or implicitly. We use r to denote 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, or a product can be rated as relevant or not relevant). The set of observed data can be viewed as a very sparse $n \times m$ matrix, \mathbf{R} , since a typical user only rates a very small fraction of the items. In the matrix \mathbf{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 not voted

¹http://www.movielens.org

on by the user. For example, let us consider the following matrix where the rows represent the users, columns are used to represent items and r = 2.

U	I_1	I_2	I_3	I_4	I_5		
U_1	2	2	0	1	0	•	
U_2	0	0	1	2	0	•	
U_3	2	2	0	0	0	•	
U_4	2	1	0	1	0	•	
U_5	0	0	0	0	2	•	
	•	•	•	•	•	•	

Table 1. Database of user rates. R.

3.1**BN-based Collaborative RS**

A Bayesian network is a directed acyclic graph (DAG), where the nodes represent the variables from the problem we want to solve. In this kind of graph, knowledge is represented in two ways [20]: qualitatively, showing the (in)dependencies between the variables, and quantitatively, expressing the strength with which we believe the dependence relationships, and represented using a set of conditional probability distributions. When we are interested in representing our knowledge by means of Bayesian networks, the first task is to select those variables which are relevant to the problem we are tackling. Each variable will be a node in the DAG and whenever two variables are related, a path must exist between them in the graph. These connections can be determined from an input dataset by means of a learning algorithm.

The objective of this section is to model the user ratings. In order to represent relationships between users, \mathcal{U} , there might be a set of nodes in the BN which match the set of users. Each user variable U_a will therefore represent the probability distribution associated to its pattern of rating, i.e. information about the probability that U_a could vote with value *i*, $Pr(U_a = i)$, with $i \in \{1, 2, ..., r\}$.

BN learning implies searching over different network structures, and is exponential in the number of variables. We must therefore impose restrictions on the way the variables are related (the links in the graph) in order to make this stage efficient. In a collaborative RS, the vote prediction for a given user depends on the votes of the people with similar tastes or preferences. In order to facilitate the presence of these relationships in the model, we therefore propose that a new of set of nodes \mathcal{V} be included to denote collaborative votes. There is one collaborative node for each user in the system, i.e. $\mathcal{V} = \{V_1, V_2, \dots, V_n\}$. These nodes will also be used to estimate the probability distributions of the user votes and they will therefore take their values in the same domain as \mathcal{U} , i.e. $\{1, 2, \ldots, r\}$.



Figure 1. Collaborative Recommending System Topology.

3.2Learning stage

The parent set of a variable V_a in the graph, $Pa(V_a)$, will be learned from the database of votes, 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. Given a similarity measure, the set $Pa(V_a)$ can therefore be obtained by using a threshold or considering only the first p variables in the ranking.

A first idea would be to select those users with the highest absolute value of Pearson's correlation coefficient (PCC)² to measure the similarity between users, a criterion which is normally used as the basis for calculating the weights in different collaborative systems.

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

where the summations over j are over those items for which users U_a and U_b have recorded votes. If there are no common items in U_a and U_b voting records, then $PCC(U_a, U_b) = 0$ by default. Additionally, \overline{r}_a is the mean vote for user U_a , i.e. $\overline{r}_a = \frac{1}{|Pa(U_a)|} \sum_{I_k \in Pa(U_a)} r_{a,k}$.

From our point of view, various problems arise when this correlation coefficient is applied on our domain, where the dataset R is a very sparse matrix. For example, considering the data in Table 1, $PCC(U_1, U_2) = -1$ since both users only rated one item in common. In this case, U_1 will be set as a parent of V_2 and also U_2 is a parent of V_1 , resulting in low quality parent sets.

In order to avoid this situation, we propose that a different, but complementary, criterion be used and which takes into account the number of items that both U_a and U_b rated simultaneously, i.e. their overlap degree. It should be noted that we are not considering the particular votes, merely the fact that the users rated an item or not. We are considering that the greater the probability of a user U_a rating an item which has been also rated by U_b , the higher the quality of U_b as the parent of variable U_a , and the criterion can therefore

²The use of the cosine measure has also been explored, but without improving the performance of the system.

be defined by

$$D(U_a, U_b) = \frac{|I(U_a) \cap I(U_b)|}{|I(U_b)|}$$

It should be noted that with this criterion, we are considering that the greater the number of items in common for U_a and U_b , the greater the similarity values. In order to illustrate how this measure works, we will consider the following example: let us consider a situation where U_a has rated 100 movies and U_b has rated 15 movies, 12 of these in common with U_a . In this case, although U_b could be considered as a more selective user than U_a , knowing that U_b rated a movie could be a good approximation about the probability that U_a will rate the same movie, $D(U_a, U_b) = 0.8$. On the other hand, if we know that user U_a rated a movie, we are not so confident about the possibility of U_b also voting for this movie, $D(U_b, U_a) = 0.12$.

The final similarity measure that we propose is therefore a combination of both criteria: vote correlation between common items and the overlap degree, i.e.

$$sim(U_a, U_b) = D(U_a, U_b) \times abs(PCC(U_a, U_b))$$
(2)

where *abs* denotes the absolute value. It should be noted that with this measure we are considering that the vote of both similar users and users with opposite tastes helps in the prediction of the final vote for an active user.

3.3 Estimating the conditional probability distributions

In order to complete the model specification, the numerical values for the conditional probabilities must be estimated from the datasets, but prior to this, we will introduce some notation. Thus, given a variable X_i , lowercase letters are used to denote the variable realization. For instance, $x_{i,j}$ denotes the fact 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.

We must distinguish between user, \mathcal{U} , and collaborative nodes, \mathcal{V} . The first ones, since they are root nodes in the graph, store marginal probability distributions, with size lineal with the number of states (O(r)), whereas the second ones must store a set of conditional probability distributions with size exponential with the number of parents m $(O(r^m))$. Since a collaborative node might be related with a great number of users, the assessment and storage of these probability values can be quite complex, and so we propose that a canonical model, similar to the one presented in [5], be used to represent the conditional probabilities, and this will allow us to design a very efficient inference procedure. For a given node X_i , we must therefore define these probabilities as follows:

$$Pr(x_{i,j}|pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j})$$
(3)

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

- For every user node U_k , we need to assess the a priori probability distribution over his/her voting pattern. In this paper, we propose $Pr(u_{k,s}) = N(u_{k,s})/|I(U_k)|$, with $|I(U_k)|$ being the number of products that user U_k has rated in the dataset, and $N(u_{k,s})$ being the number of times that user U_k rated a product with the value s.
- Focusing on collaborative nodes V, in order to estimate the final vote for the user V_a we must compute those weights given by users U_b with similar tastes, i.e. U_b ∈ Pa(V_a). We propose that the following weighting scheme be used

$$w(u_{b,t}, v_{a,s}) = \frac{1}{|Pa(V_a)|} \frac{N^*(u_{b,t}, v_{a,s}) + \beta q_s}{N^*(u_{b,t}) + \beta}, 1 \le t, s \le r$$
(4)

where the value $N^*(u_{b,t}, v_{a,s})$ is the number of items from the set $I(U_a) \cap I(U_b)$ that having been voted with value t by user U_b have also been voted with value s by user U_a , and $N^*(u_{b,t})$ is the number of items in $I(U_a) \cap I(U_b)$ voted with value t by user U_b . Values β and q_s are the parameters of a Dirichlet prior over user ratings with $\sum_{i=1}^r q_i = 1$.

4 Adding Group Layer

As groups are usually created by their members, we will not consider questions about how groups are formed nor how they are managed. We will therefore assume that we know the composition of the groups, and our problem is to study how this information can be represented in the BN and also how to form recommendations for groups, i.e. how the inference processes can be performed.

Following [19], there are two possible alternatives for group settings: the first one is to create a "pseudo-user" that represents the group tastes, and produces the group's recommendation in the same way as we obtain a single user's recommendation. The second alternative is to consider a group as a new entity where recommendations are made by considering the particular recommendations of its members in some way. This approach has the additional advantage of allowing a better explanation of the given recommendations. In this paper, we will consider this second approach since it better retains the essence of the BN philosophy.



Figure 2. Modeling groups

For each group in the system, we will therefore include a new node in the BN with the set of collaborative nodes corresponding to the group members as parents. Figure 2 illustrates a group G_a with users U_1, U_2 and U_3 as its members. We represent that the prediction of the group vote depends on the votes predicted by the collaborative component for each of its members. We use dashed lines to represent users-group relations since we might assume that the group is formed explicitly.

It is interesting to note that by modeling a group with a BN, the recommended vote for G_a will not be obtained by combining the group members' ratings in an isolated way. The predicted rate will instead depend on all the possible vote configurations for the group's members. For example, in a two-member group (with two alternative ratings) we consider on one side the likelihood of both V_1 and V_2 voting with value 1, $Pr(v_{1,1}, v_{2,1})$, and on the other, the probability, under this circumstance, of the group voting with value s, $Pr(G_a = s|v_{1,1}, v_{2,1})$. This pair of probabilities will be computed for all the possible alternative ratings for the group's members, i.e. $\{v_{1,2}, v_{2,1}\}, \{v_{1,1}, v_{2,2}\}$ and $\{v_{1,2}, v_{2,2}\}$. The final rate is then obtained by combining this rating, i.e.

$$Pr(G_a = s) = \sum_{i,j} Pr(G_a = s | v_{1,i}, v_{2,j}) \times Pr(v_{1,i}, v_{2,j}).$$

4.1 Probabilities at group nodes

The objective of this section is to consider how the conditional probability distributions for group nodes can be assessed. It should be noted that for each group G_i , we must assess the values $P(G_i = k|pa(G_i)), k = 1, ..., r$, which represent the probability of the group voting with the value k given that we know a particular configuration for its members. This conditional probability distributions can be considered as a "social value function" describing how the opinions of the members affect the group's recommendation. Since the group might be large, we will use canonical models to define these probabilities. Following the ideas in [19], we will consider three alternatives:

1. Average: Group vote (happiness) can be considered

Table 2. Representing social value functions by means of conditional probability distributions

Prob.	AVG	MAX
$P(G_i = k pa(G_i))$	$k = \{1, 2, 3\}$	$k = \{1, 2, 3\}$
$P(G_i 1, 1, 1)$	$\{1, 0, 0\}$	$\{1, 0, 0\}$
$P(G_i 1, 1, 2)$	$\{0.66, 0.33, 0\}$	$\{0, 1, 0\}$
$P(G_i 1, 1, 3)$	$\{0.66, 0, 0.33\}$	$\{0, 0, 1\}$
$P(G_i 1, 2, 1)$	$\{0.66, 0.33, 0\}$	$\{0, 1, 0\}$
$P(G_i 1, 2, 2)$	$\{0.33, 0.66, 0\}$	$\{0, 1, 0\}$
$P(G_i 1, 2, 3)$	$\{0.33, 0.33, 0.33\}$	$\{0, 0, 1\}$
$P(G_i 1, 3, 1)$	$\{0.66, 0, 0.33\}$	$\{0, 0, 1\}$
$P(G_i 1, 3, 2)$	$\{0.33, 0.33, 0.33\}$	$\{0, 0, 1\}$
$P(G_i 1, 3, 3)$	$\{0.33, 0, 0.66\}$	$\{0, 0, 1\}$
$P(G_i 3, 3, 3)$	$\{0, 0, 1\}$	$\{0, 0, 1\}$

as the average vote (happiness) of its members. This situation can be represented by means of a canonical model such as the one in Equation 3, i.e

$$P(G_i = k | pa(G_i)) = \sum_{j=1}^{|Pa(G_i)|} w(G_{i,k}, V_{j,s})$$

where $w(G_{i,k}, V_{j,s})$ is the weight (effect) that the j^{th} group member voting the s^{th} value has in the vote k of the group. These weights can be defined as:

$$w(G_{i,k}, V_{j,s}) = \begin{cases} \frac{1}{|Pa(G_i)|} & \text{if } k = s, \\ 0 & \text{otherwise.} \end{cases}$$

It should be noted that in this case we do not consider situations where there are users with a high quality opinion (experts). Nevertheless, these could easily be taken into account by means of a proper modification of the weights. The second column of Table 2 illustrates the performance of the average criterion for a group G_i with three members voting in the range 1 to 3.

Maximum: the group vote is equal to the vote of the happiest group member. The conditional probability distributions are therefore defined as

$$P(G_i = k | pa(G_i)) = \begin{cases} 1 & \text{if } k = \max\{pa(G_i)\}\\ 0 & \text{otherwise} \end{cases}$$

with $\max\{pa(G_i)\}$ being the maximum vote in the configuration $pa(G_i)$. For example, let us consider the probabilities in Table 2, third column.

3. Minimum: the group vote is associated to the happiness of the least happy members, i.e. the minimum

vote of the individuals. This situation can be represented by means of

$$P(G_i = k | pa(G_i)) = \begin{cases} 1 & \text{if } k = \min\{pa(G_i)\} \\ 0 & \text{otherwise} \end{cases}$$

with $\min\{pa(G_i)\}$ being the minimum vote in the configuration $pa(G_i)$.

5 Computing the recommendations: Inference

Once the BN is completed, it specifies a complete joint probability distribution over all the variables, i.e. for a given configuration $c = (x_1, x_2, \ldots, x_n)$ over the set of variables X_1, \ldots, X_n , with x_i being the value that variable X_i takes in c then:

$$Pr(c) = \prod_{i=1}^{n} Pr(x_i | pa(x_i))$$

with $pa(x_i)$ being the values that the parent set of X_i in the BN, $Pa(X_i)$, takes in c.

Given this joint distribution, we can answer all possible inference queries by marginalization (summing out over irrelevant variables). In a BN framework, the user usually has some evidence about the state that a variable (or a set of variables) takes, and the problem is to compute the conditional (a posteriori) probability distribution for a variable given the evidence, $Pr(X_i|ev)$. Although general purpose algorithms exist, this process takes exponential time with the number of nodes in the graph when applied to a BN with the proposed topology.

In order to present inference methods which are specially designed to work with our model, it is previously necessary to discuss how the users should interact with the system. In our case, the objective is to predict the vote that a group of users, G_a , should give for a set of items, \mathcal{I}^3 . In this case, we could consider that the set of unobserved items acts as evidence, i.e. $ev = \mathcal{I}$, and we must compute $Pr(G_a = s|ev)$, $1 \leq s \leq r$. Let us examine how to include this evidence in the BN.

Given the evidence \mathcal{I} , we are focusing on a world where the items in \mathcal{I} are relevant. It therefore seems natural that we must change our belief in the vote distribution for any user who previously rated an item in \mathcal{I} . For example, consider that item I_j acts as evidence, i.e $ev = \{I_j\}$. Let U_i be a user who voted item I_j with value s. In this case, given the evidence, it seems reasonable to increase our belief about the fact that "variable U_i takes the state s". The particular way in which the belief over the candidate rates changes could be captured by means of the following expression:

$$Pr(U_i = s | ev) = \begin{cases} \frac{N_{ev}(u_{i,s})}{N_{ev}(U_i)} & \text{if } N_{ev}(U_i) > 0\\ Pr(U_i = s) & \text{otherwise} \end{cases}$$
(5)

with $N_{ev}(U_i)$ being the number of items in the evidence set voted by user U_i with value s and $N_{ev}(U_i)$ the number of items in the evidence set voted by user U_i . It should be noted that when there is no evidence for a given user, the a posteriori probability coincides with the a priori probability distribution. In order to illustrate this situation, let us consider an RS where the items are a set of movies to be rated with 1 to 3 stars, and a user U who awarded "The Lion King (LK)" and "Finding Nemo (FN)" 2 and 3 stars, respectively. Given the evidence sets $\mathcal{I}_1 = \{LK\}$ and $\mathcal{I}_2 = \{LK, FN\}$, the a posteriori probabilities are: $Pr(U = 2|\mathcal{I}_1) = 1$ and $Pr(U = 1|\mathcal{I}_1) = Pr(U = 3|\mathcal{I}_1) = 0$ whereas $Pr(U = 2|\mathcal{I}_2) = 0$.

5.1 Evidence Propagation

Considering the typical behavior of a collaborative system, we must predict the rate that an active group G_a should assign to the evidence set. From a probabilistic point of view, we must compute $Pr(G_a = s|ev), 1 \le s \le r$, i.e.

$$Pr(G_a = s|ev) = \sum_{c \in \mathcal{U}, V, G} Pr(g_{a,s}, c|ev).$$
(6)

where we sum over all the possible configurations c for the set of irrelevant variables in \mathcal{U}, V, G^4 .

These probability values can be computed by propagating the evidence through the network. If we consider that in a BN a node is independent of all its predecessors since we know the particular values that its parents can take and also that by using the advantages of the canonical models used, these final probabilities could be computed efficiently. Nevertheless, we must distinguish two different situations: the first where the canonical weighted-sums (Equation 3) are used to represent all the conditional probability distributions, and the second where we use max or min gates to represent the probability distributions for group nodes.

5.1.1 Propagation with canonical weighted-sum

In this case, the probabilities can be obtained by means of a straight application of the following theorem [7] that shows how we can compute the exact probability values when using canonical weighted-sum representations.

Theorem 1: Let X_a be a node in the collaborative BN network, m_{X_a} be the number of parents of X_a , Y_j be a node

 $^{^{3}\}text{In}$ our case we assume that no member of the group has rated an item in $\mathcal{I}.$

⁴It should be noted that our model can also be used to *find good items*. In this case, we could recommend to the group G_a those items with the highest probabilities of being liked. This can be done by instantiating in turn all the unseen items for G_a .

in $Pa(X_a)$ and l_{Y_j} the number of states that Y_j takes. If the conditional probability distributions can be expressed under the conditions given by Equation 3, then the exact a posteriori probabilities can be computed using the following formula:

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

In this case, a top-down propagation mechanism can be designed, where the posterior probabilities of one layer are obtained using the posterior probabilities computed in the previous layer, starting from the user nodes. It should be noted that for the user nodes, $Pr(U_i|ev)$ is computed using equation 5.

5.1.2 Propagation with MIN and MAX gates

In this case, using the independencies in the BN and the fact that the evidence only contributes in user nodes,

$$Pr(G_a = s|ev) = \sum_{pa(G_a)} Pr(g_{a,s}|pa(G_a))Pr(pa(G_a)|ev))$$

where the sum is over all the possible configuration of the parent set of $G_a, Pa(G_a)$, in the graph.

We therefore have two different problems:

Computing Pr(pa(G_a)|ev)): This is quite expensive since for each possible configuration in the set Pa(G_a), it is necessary to sum over a large set of variables, which is doubly exponential,

$$Pr(pa(G_a)|ev) = \sum_{c \in \mathcal{U}, V} Pr(pa(G_a), c|ev)$$

In order to solve this problem, we therefore propose that this probability value be approximated by assuming that the a posteriori probability for the collaborative nodes is independent given the evidence, i.e.

$$Pr(pa(G_a)|ev)) = \prod_{i=1}^{|Pa(G_a)|} Pr(v_{i,j}|ev)$$

with j being the value that the collaborative node V_i takes in the configuration $pa(G_a)$.

2. Once these values have been computed we still need to combine them with $Pr(g_{a,s}|pa(G_a))$ in order to obtain the final probability distributions. This combination requires summing over the configurations in $Pa(G_a)$, which is exponential in the number of group members, $O(r^m)$.

It can be proved that under the above independence assumption, the probability distribution $Pr(G_a|ev)$ can be computed easily and efficiently (in an order linear with the number of the group members).

We shall first introduce some notation. Let s_1, s_2, \ldots, s_n be an order over the states that variable X takes such that $s_i < s_{i+1}, i = 1, \ldots, n-1$, let S_X be a set of states in X and $Pr(X \in S_x)$ the probability that variable X takes any state in S_x , i.e. if $S_x = \{s_1, s_2, s_3\}$ then $Pr(X \in S_x) =$ $Pr(s_1)+Pr(s_2)+Pr(s_3)$. In addition, let m be the number of members of the group G_a , i.e. m = |Pa(G)|,

We shall now look at how to compute the a posteriori values for min and max gates:

• Min - Gate: We must consider that $S_n \subset S_{n-1} \subset \ldots \subset S_2 \subset S_1$ being $S_i = \{s_i, s_{i+1}, \ldots s_n\}$.

-
$$Pr(G_a = s_n|ev)$$
 is equal to

$$\prod_{i=1}^m Pr(V_i \in S_n|ev) = \prod_{i=1}^m Pr(v_{i,n}|ev)$$
- $Pr(G_a = s_k|ev)$, for $k = 2, \dots, n-1$, is equal to
 $\left(\prod_{i=1}^m Pr(V_i \in S_k|ev)\right) - Pr(G_a \in S_{k+1}|ev).$

- $Pr(G_a = s_1 | ev)$ is defined as $1 - Pr(G_a \in S_2)$.

 Max - Gate: the situation is similar to the previous one, but considering that S₁ ⊂ S₂ ⊂ ... ⊂ S_{n-1} ⊂ S_n being S_i = {s₁,..., s_{i-1}, s_i}. Therefore,

$$Pr(G_a = s_1|ev)$$
 is equal to

$$\prod_{i=1}^m Pr(V_i \in S_1|ev) = \prod_{i=1}^m Pr(v_{i,1}|ev)$$
 $Pr(G_a = s_k|ev)$, for $k = 2, \dots, n-1$, is equal to

$$\left(\prod_{i=1}^{m} \Pr(V_i \in S_k | ev)\right) - \Pr(G_a \in S_{k-1} | ev)$$
$$\Pr(G_a = s_n | ev) = 1 - \Pr(G_a \in S_{n-1}).$$

5.2 Vote Recommending

Once we have computed the a posteriori probability distribution $Pr(G_a|ev)$, the problem is to decide on the final vote that the system might recommend to the group. There are two possibilities that we shall study in this paper. MP This is the most naive alternative, selecting the vote with the highest a posteriori probability, i.e.

vote =
$$\arg\max\{Pr(G_a = s|ev)\}$$

For example, if $Pr(G_a = 1|ev) = 0.225$, $Pr(G_a = 2|ev) = 0.25$, $Pr(G_a = 3|ev) = 0.175$, $Pr(G_a = 4|ev) = 0.3$ and $Pr(G_a = 5|ev) = 0.05$ the system will select vote 4.

DP This alternative considers how the new evidence has been spread through the model trying to reduce the bias that, a priori, the group should have for a given vote. As we shall see, this alternative is particularly appropriate for those situations where the minimum and the maximum criteria are used to merge the individuals' votes. It should be noted that the min-gate is highly biased towards low value votes whereas the max-gate is biased towards high value rates. Therefore, the idea is to measure only the new piece of evidence that each candidate vote receives by means of the instantiation of the items. This new piece of evidence can be easily defined by considering the difference between the a priori (without evidence) and the a posteriori probability values, and returning the vote with the maximum difference.

$$vote = \arg\max\{Pr(G_a = s | ev) - Pr(G_a = s)\}$$

Following with the example above, and considering the a priori probability distribution $Pr(G_a = 1) = 0.1$, $Pr(G_a = 2) = 0.2$, $Pr(G_a = 3) = 0.25$, $Pr(G_a = 4) = 0.25$ and $Pr(G_a = 5) = 0.2$, the vote 1 will be recommended.

One extremely important aspect in RS is the ability to explain the recommendations. In our case, the system would present both, the proposed group vote and also the results that each group member obtains individually. The purpose is to show group agreement and not the particular ratings of each member (it should be noted that showing the results for individual group members in an explanation to the group could present problems of privacy [14] in some group RS applications). For instance, the system can show a picture such as the one presented in Figure 4 which presents the voting preferences (on a scale of 1 to 5) of each group member (columns), using a black and white scale, where black means the favorite choice and white represents the least probable option. It should be noted that vote r is preferable to vote s is $Pr(U_a = r|ev) > Pr(U_a = s|ev)$ using the MP criteria and $Pr(U_a = r|ev) - Pr(U_a = r) >$ $Pr(U_a = s | ev) - Pr(U_a = s)$ using DP criteria. Thus, Figure 4 illustrates two different situations where the system could recommend the same vote (vote 4) for the group, but on the left-hand side there is not much agreement between

Table 3. Groups statistics

	PCC_{g}	TD_{PCC}	#2	#3	#4	entries
UG1	0.2301	0.0977	800	393	70	6982
UG2	0.2048	0.0853	2397	2342	632	17438
UG3	0.2061	0.0832	4167	3914	963	20264
UG4	0.2140	0.0847	4488	3589	830	17702
UG5	0.2260	0.0883	4089	2717	546	14088

the group members whereas the right-hand side displays a situation where there is better agreement.



Figure 3. Explaining the recommendations

6 Empirical Analysis.

This section presents some experimental results about the performance of the system. Since we do not have any databases for groups, we decided to use MovieLens and to learn the group relationships from it. The MovieLens dataset was collected by the GroupLens Research Project at the University of Minnesota during the seven-month period between September 19th, 1997 and April 22nd, 1998 with votes ranging from 1 to 5. The dataset contains 1682 movies and there are 943 users in the dataset, containing 100,000 transactions in the scale 1 to 5. In order to perform 5-fold cross validation, we have used the data sets U1.base and U1.test through U5.base and U5.test in MovieLens which split the collection into 80% for training and 20% for testing, respectively.

The collaborative component of the system is learned from training sets. In particular, for each collaborative node V_i we look for the 10 most similar users using the similarity measure in Equation 2. It should be noted that all the conditional probability distributions stores in \mathcal{U} and \mathcal{V} nodes have been estimated from the training set as defined in Section 3.2 with parameters $\beta = 1$ and $q_i = 1/5$.

Regarding the group relationships, we have also decided to learn the groups from the training datasets. In order to create groups, we follow some conclusions about how the groups have been created in the literature [10, 19]: the majority of the group members have similar tastes and it is usual for an administrator to create smaller sized groups (usually with two, three or four members).

Table 4	Average	Experimental	Results
	Averaue	LADEIIIIEIIIAI	nesulis

÷ .							
	Base	eline	Group layer				
	%S	MAE	%S	MAE			
MAX-MP	55.38	0.503	54.41	0.553			
MAX-DP	55.32	0.510	56.38	0.488			
MIN-MP	43.26	0.733	34.92	0.980			
MIN-DP	42.83	0.754	44.34	0.692			
AVG-MP	50.82	0.563	50.15	0.565			
AVG-DP	45.26	0.624	50.05	0.571			

In our experimentation, the groups are formed as follows: we set each user as the group administrator and calculate at most ten similar users (those users who are positively correlated in the training dataset using Pearson's correlation coefficient (PCC)). We then create groups of two, three and four members. In order to illustrate the homogeneity of the groups, we show in Table 3 the mean of Pearson's correlation coefficient between the group members and the average typical deviation, columns 2 and 3, respectively. From these values, we could conclude that although the groups are formed with related users, there is some variability among members.

The group test sets have been obtained from the Movie-Lens test datasets. Whenever we found a movie which had been rated by all the group members in the U*i*.test set, i = 1, ..., 5, we inserted a new entry in its respective group test database, UG*i*.test. This entry includes the number of the group, the common movie, and all the ratings given by its members. In Table 3 we show for each test dataset UG*i*.test the group number for each size (columns 4, 5 and 6) and column 7 presents the number of entries.



Figure 4. MAE results for each test set

Having presented the experimental framework, we shall now discuss how we will evaluate the accuracy of the system. Two different measures will be considered [9]: the percentage of success (%S), which measures the frequency with which the system makes correct predictions and the mean absolute error (MAE), which measures the average absolute deviation between a predicted rate and the group's true rate. In both measures, it is necessary to know the *real rate* of the group. Since we do not know this vote, it will be computed by means of a merging strategy using the individual real votes from UG_i.test. Three different alternatives are considered: the final group rate is the minimum (MIN), maximum (MAX), and average (AVG) rate of its members.

In order to evaluate the quality of our approach, we consider two different situations, depending on whether or not we consider the group layer in the BN. The idea is to test the benefits obtained by using MIN, MAX and AVG gates to represent social value functions in the BN-based model. The group rate in the second alternative, which could be considered as the baseline, is obtained by merging, with the MIN, MAX or AVG criteria, the rates that individually the collaborative component proposes for each group member, i.e. using the results at nodes \mathcal{V} . In addition, at nodes \mathcal{V} we consider the same two strategies for deciding the final rate: select the most probable (MP) rate or the rate with the highest difference between the a priori and the a posteriori probability distributions (DP).

Table 4 presents the average results obtained after repeating the experiment with each training and test set. It is worth mentioning that although we present the average results, the conclusions are comparable in the five experimental test sets (see Figure 4 showing the MAE results when using the MIN, MAX and AVG criteria). From this table, we can conclude that the best results for MAX and MIN value functions have been obtained when modeling the relations between the group members by means of BN using DP as the selection criteria. We think that this is because when MP criteria are used, and due to the extreme behavior of these gates, the a priori probability values have a strong weight in the final decisions. In the AVG case, the results are quite similar to those obtained with the baseline (MAE 0.565 vs. 0.563, respectively). It is interesting to note that when the group layer with AVG function is considered, there are no great differences between the use of DP or MP, although MP seems to be preferable in all the test sets. Finally, with respect to the baseline model, it is always better to use the MP criteria to decide the final vote for each user.

7 Conclusions

A general BN-based model for group recommending has been proposed in this paper, and this is an intuitive representation of the relationships between users and groups. The mechanisms used by the group to decide the proposed rate are encoded by means of conditional probability distributions stored at the group nodes. Taking into account efficiency considerations, these distributions have been assessed by means of canonical modes. The use of these models also allows the a posteriori probabilities to be computed (necessary for deciding the recommended rate) in linear time. Guidelines for how to estimate the probability values from a data set and also how the RS interacts with the users have been given. It must be noted that the proposed model is quite general, since it can be applied to different recommendation tasks (such as *find good items* or *predict rates*) for a single item or for a set of items.

By way of future work, we are planning to evaluate the model with real data, involving real groups to determine the quality of the recommendations provided. We will also attempt to incorporate mechanisms that enable consensus to be reached between group members and specifications for other social value functions to be incorporated (such as the majority rate).

Acknowledgments:

This work has been supported by the Spanish 'Ministerio de Educación y Ciencia' and 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía'under Projects TIN2005-02516 and TIC-276, respectively.

References

- M. Balabanovi and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.
- [2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [3] C. Butz. Exploiting contextual independencies in web search and user profiling. In *Proc. of World Congress on Computational Intelligence*, pages 1051–1056, 2002.
- [4] L. M. de Campos, J. M. Fernández-Luna, M. Gómez, and J. F. Huete. A decision-based approach for recommending in hierarchical domains. In *ECSQARU*, pages 123–135, 2005.
- [5] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. Generalizing e-bay.net: An approach to recommendation based on probabilistic computing. In *1er. Workshop on Web Personalization, Recommender Systems and Int. User Interface*, pages 24–33, 2005.
- [6] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. A bayesian network approach to hybrid recommending systems. In *Int. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 2158–2165, 2006.
- [7] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. Modeling canonical weigted-sum. In *Technical Repot*, 2007.
- [8] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. J. Mach. Learn. Res., 1:49–75, 2001.
- [9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., 22(1):5–53, 2004.

- [10] A. Jameson. More than the sum of its members: challenges for group recommender systems. In AVI '04: Proceedings of the working conference on Advanced visual interfaces, pages 48–54, New York, NY, USA, 2004. ACM Press.
- [11] S. Kangas. Collaborative filtering and recommendation systems. In VTT Information Technology, 2002.
- [12] F. J. Martin and M. Torrens. Partystrands: Group recommendations. In *Recommenders06: The present and future of recommenders*, 2006.
- [13] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. User Model User-Adap Inter., 14:37–85, 2004.
- [14] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Model User-Adap Inter., 16:281– 319, 2006.
- [15] J. E. McCarthy and T. D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work, page 348, New York, NY, USA, 2000. ACM Press.
- [16] K. McCarthy, M. Salam, L. Coyle, L. McGinty, B. Smyth, and P. Nixon. Group recommender systems: a critiquing based approach. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 267–269, New York, NY, USA, 2006. ACM Press.
- [17] K. Miyahara and M. J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *Pacific Rim International Conference on Artificial Intelligence*, pages 679–689, 2000.
- [18] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, New York, NY, USA, 2000. ACM Press.
- [19] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of user. In *ECSCW*, pages 199–218, 2001.
- [20] J. Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
- [21] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In 17th Conference on Uncertainty in Artificial Intelligence, pages 437– 444, Seattle, Washington, August 2–5 2001.
- [22] P. Resnick and H. R. Varian. Recommender systems. Commun. ACM, 40(3):56–58, 1997.
- [23] V. Robles, P. Larrañaga, J. Peña, O. Marbán, J. Crespo, and M. Pérez. Collaborative filtering using interval estimation naive bayes. In *Lecture Notes in Artificial Intelligence*, pages 46–53, 2003.
- [24] S. N. Schiaffino and A. Amandi. User profiling with casebased reasoning and bayesian networks. In *IBERAMIA-SBIA 2000 Open Discussion Track*, pages 12–21, 2000.
- [25] Z. Yu, X. Zhou, Y. Hao, and J. Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Modeling and User-Adapted Interaction*, 16(1):63–82, 2006.