User Model User-Adap Inter (2009) 19:207–242
DOI 10.1007/s11257-008-9061-1

ORIGINAL PAPER

Managing uncertainty in group recommending processes

Luis M. de Campos · Juan M. Fernández-Luna · Juan F. Huete · Miguel A. Rueda-Morales

Received: 25 February 2008 / Accepted in revised form : 26 October 2008 /
Published online: 21 November 2008

Abstract While the problem of building recommender systems has attracted considerable attention in recent years, most recommender systems are designed for recommending items to individuals. The aim of this paper is to automatically recommend a ranked list of new items to a group of users. We will investigate the value of using Bayesian networks to represent the different uncertainties involved in a group recommending process, i.e. those uncertainties related to mechanisms that govern the interactions between group members and the processes leading to the final choice or recommendation. We will also show how the most common aggregation strategies might be encoded using a Bayesian network formalism. The proposed model can be considered as a collaborative Bayesian network-based group recommender system, where group ratings are computed from the past voting patterns of other users with similar tastes.

Keywords Group recommending · Management of uncertainty · Probabilistic graphical models

L. M. de Campos · J. M. Fernández-Luna · J. F. Huete (✉) · M. A. Rueda-Morales
Department of Computer Science and Artificial Intelligence, University of Granada,
18071 Granada, Spain
e-mail: jhg@decsai.ugr.es

L. M. de Campos
e-mail: lci@decsai.ugr.es

J. M. Fernández-Luna
e-mail: jmfluna@decsai.ugr.es

M. A. Rueda-Morales
e-mail: mrueda@decsai.ugr.es

1 Introduction

Recommender systems (RS) provide specific suggestions about items (or actions) within a given domain which may be considered of interest to the user (Resnick and Varian 1997). Depending on the information used when recommending, traditional RS are mainly classified into content and collaborative-based RS, although hybrid approaches do exist. The first type recommends a product by considering its content similarity with those products in which the user has previously expressed an interest. The second alternative attempts to identify groups of people with similar tastes to the user and to recommend items that they have liked. Most RS are designed for individual use, i.e. there is an active user who receives recommendations about certain products once they have logged on to the system.

In this paper, we will focus on the related problem of *group recommending* (*GR*), where the objective is to obtain recommendations for groups of people (Jameson and Smyth 2007). This kind of RS is appropriate for domains where a group of people participate in a single activity such as watching a movie or going on vacation and also in situations where a single person must make a decision on behalf of the group.

In one way or another, GR involves merging different individual preferences. In these situations, it is natural that one of the most important issues is the search for an aggregation mechanism to obtain recommendations for the group. According to Pennock and Wellman (2005) "... *there is nothing close to a single well-accepted normative basis for group beliefs, group preferences or group decision making.*", and many aggregation strategies can therefore be found in literature for group decisions (Masthoff 2004; Masthoff and Gatt 2006; Yu et al. 2006; Jameson and Smyth 2007). It is typically assumed that member preferences are given using a rating domain (let us say from 5*, *really like*, to 1*, *really hate*). An aggregation strategy is then used to determine the group rating. For example, let us consider a group with three individuals, *John, Ann* and *Mary*, where *John* rates a product 5*, *Ann* rates it 2*, and *Mary* rates it 5*. Following an average aggregation criterion, we could then say that the group rating for this product is 4*.

As in the previous example, the methods proposed in GR literature (see Jameson and Smyth 2007 for a review) do not deal with uncertainty. They assume that the inputs of the aggregation functions (i.e. user preferences) are precise and use a merging strategy to compute precise outputs. This assumption is not necessarily true, especially if we consider that the user's preferences are normally determined by means of automatic mechanisms. In these cases, a probability distribution over the candidate ratings might be used to express user likelihoods. For example, Table 1 shows the probability distributions representing the preferences of three users (A, B, and C). In this case,

Table 1 User ratings for a given item

User	1*	2*	3*	4*	5*
A	0.2	0.2	0.2	0.19	0.21
B	0	0	0	0.1	0.9
C	0.49	0	0	0	0.51

although 5* might be considered the most probable rating, we will not have the same confidence about every situation.

Surprisingly, little attention has been paid in GR literature to the problem of managing uncertainty although it has been well established in the general group decision framework (see Clemen and Winkler 1999; Genest and Zidek 1986 for a review). In this paper, therefore, we will focus on this particular problem. We maintain that two different sources of uncertainty can be found in group recommending processes: the uncertainty shown when user preferences are set, i.e. the user's personal opinion about an item or feature; and the uncertainty which is inherent to the merging process.

The purpose of this paper is to investigate the value of using Bayesian networks (BN) to represent how different individuals in a group interact in order to make a final choice or recommendation. In our approach, the BN formalism is used to represent both the interactions between group members and the processes leading to the final choice or recommendation. We will show how common decision rules in literature could be managed by adequately designing canonical models with the BN language, thereby shedding new light on the combination processes. Discussion about subjects such as how the groups are formed, how long they have existed, relationships between group members, how the group might interact to reach a consensus, etc. are beyond the scope of this paper. We shall assume that all the individuals use the same set of labels to express their preferences for an item, and that these preferences are represented by means of a probability distribution (probably estimated from a data set).

We consider BNs appropriate because they combine a qualitative representation of the problem through an explicit representation of the dependence relationships between items, users and groups, with a quantitative representation by means of a set of probability distributions to measure 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 represent two challenges.

The second section of this paper briefly examines group recommender systems and related work. Section 3 presents the proposed BN-based model which enables the interaction between individuals to be represented. Section 4 examines how to represent the strength of the individuals' interactions (i.e. conditional probability distributions) and Sect. 5 discusses how inference is performed in order to make recommendations to the group. Section 6 examines the experimental framework. Section 7 discusses the experimental results obtained when considering uncertainty in individual ratings and in Sect. 8 we study those situations where the process behind the group rating is also uncertain. Finally, our conclusions and comments regarding further research are discussed in Sect. 9.

2 Classification of group recommender systems and related work

Although GR is quite a new research topic, many papers on this problem have already been published. The specific objectives of recommender systems in the research published so far are determined by the characteristics of the domain for which the system has been developed. These characteristics significantly affect the choice of design and

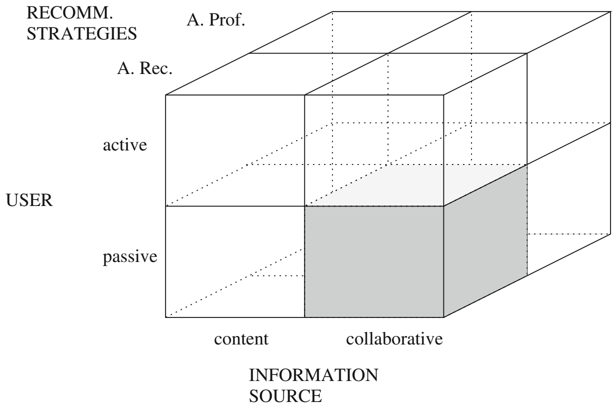


Fig. 1 Classification of Group Recommending Systems

each publication therefore focuses on a specific issue (from how to acquire information about group preferences or how the system generates and explains the recommendations to studying the mechanism used to reach a consensus (Jameson and Smyth 2007)). As a result, relating the different approaches is a difficult task.

In this section, we will present a new classification taxonomy for group recommending systems. This classification is based on three independent components of primary importance in the design of a group recommending system and not on the particular techniques used to solve each problem: the information source, the aggregation criterion used to make the recommendations, and the user's interaction with the system. Figure 1 shows a graphical representation of the proposed classification.

– *Source of information*: This classification criterion, which has been borrowed from classical RS literature (Adomavicius and Tuzhilin 2005), distinguishes between *content-based* (CB) and *collaborative filtering* (CF). In the first case, the recommended items are those which are similar to the ones that individuals have found interesting in the past. As a result, it is necessary to analyze the content's features for recommending.
  The second alternative considers that the recommendations for a target product have been obtained by considering how people with similar tastes rated a product in the past. These systems are based on the idea that people will agree in future evaluations if they have also agreed in their past evaluations. The information sources are therefore the preference ratings given by similar users.
  A new category can obviously be obtained if we consider hybrid approaches that combine both (collaborative and content-based) methods.[1]
– *Recommendation strategies*:
  Once we have the information to hand, the strategy used for aggregating this information is a central point in group recommending, and generally in any group decision process. In this case, two different approaches can be distinguished. The first

[1] Without loss of generality, we have decided not to include this category in our taxonomy since, to the best of our knowledge, no study has tried to combine both techniques in the group recommending framework.

approach, *aggregating recommendations* (AR), is a two-step strategy, where an individual recommendation is first obtained for each group member, and then a common recommendation is obtained by merging these individual recommendations. In the second approach, *aggregating profiles* (AP), the objective is to obtain a common profile by representing group preferences. This can be done explicitly, where the individuals use a common group account to give their preferences, or implicitly, by means of an aggregation mechanism for the different individuals' profiles or preferences.

– *Individual interactions*
  Finally, a group recommending system can also be categorized by considering the way in which the users interact with the system. The individuals can be dichotomized into *passive members* (PM) and *active members* (AM). Focusing on the *active* members, the final purpose is to reach a consensus between the group members and, like many decision support system approaches, it is necessary for the users to evaluate the system recommendations. In contrast, when the members are *passive*, the final purpose is only to provide a recommendation to the group, as might be the case when using an RS in a marketing campaign. In this situation, the individuals do not interact with the system in order to evaluate the proposed recommendations.

Since we use three non-overlapping criteria for classification purposes, a given GRS can be classified using three labels, one for each category. For instance, a GRS can be classified as CB+AP+PM if the group profile is obtained by combining the information about the content of the items which have been previously evaluated by each user. This profile will be used to send the final recommendations to the group.

2.1 Related work

Once the taxonomy has been presented, we will then go on to classify previously published GR systems.

– CB+AP+PM: most published GRSs might be included in this category. For example, let us consider MusicFX (McCarthy and Anagnost 2000). Given a database of member preferences for musical genres (each user rates each of the 91 genres on a five-point scale), the group profile is computed by summing the squared individual preferences. Using a weighted random selection operator, the next music station to be played is then selected. No interaction with the system is possible except by changing user preferences.
  The inputs in the case of group modeling (Masthoff 2004) are user preferences (ratings) for a series of programs, and in this paper we study the performance of several aggregation strategies. The article (Yu et al. 2006) presents various TV program recommendations for multiple viewers by merging individual user preferences on features (e.g. genre, actor, etc.) to construct a group profile. The aim of the aggregation strategy is to minimize the total distance in such a way that the merged profile is close to most user preferences, thereby satisfying most of the group.
– CB+AP+AM: The Travel Decision Forum (Jameson 2004) was developed to help a group of users agree on the desired attributes of a vacation. This system allows

group members to collaboratively specify their preferences and to reach an agreement about an overall solution. In this case, a group profile is obtained through the interaction of the members, taking into account the system's current recommendation which is obtained by aggregating individual preferences for each dimension. In the article (Kudenko et al. 2003), a system is presented to help a group of users reach a joint decision based on individual user preferences.

- CB+AR+PM: Intrigue (Ardissono et al. 2003) recommends tourist attractions for heterogeneous groups that include homogeneous subgroups where the members have similar preferences. In this system, the users record their preferences for a series of tourist attractions, and recommendations (obtained using a fuzzy AND) are then merged using a weighted scheme where each weight represents the relevance of the corresponding subgroup (for instance, a subgroup could be particularly influential since it represents a large portion of the group). Although the system explains their recommendations, it has no means of interacting with the user.
- CF+AR+PM: Polylens (O'Connor et al. 2001), an extension of MovieLens (Herlocker et al. 2004), recommends movies to groups of users. This system uses a nearest neighbor-based algorithm to find the individuals with the most similar tastes to those of each group member and to obtain recommendations for every user. The voting preferences of these individuals are then merged according to the principle of least misery (minimum criterion). Under the same classification, (Chen et al. 2008) uses genetic algorithms to learn the group rating for an item that best fits the existing ratings for the item given by the individuals and the subgroups. The idea is that it is possible to learn how the user interacts from the known group ratings. The proposed algorithm therefore recommends items based on the group's previous ratings for similar items.

2.1.1 The role of uncertainty

As far as the authors are aware, the role of uncertainty in group recommending processes has not been considered. Nevertheless, many papers have been published which tackle this problem when recommendations are made to individual users (Zukerman and Albrecht 2001; Albrecht and Zukerman 2007). Focusing on probabilistic approaches, those relating to the one presented in this paper include *content-based RSs* (Mooney and Roy 2000; de Campos et al. 2005), *collaborative filtering RSs* (Breese et al. 1998; Schiaffino and Amandi 2000; Butz 2002; Lekakos and Giaglis 2007; Miyahara and Pazzani 2000; Heckerman et al. 2001) and *hybrid* methods (Popescu et al. 2001; de Campos et al. 2006).

In terms of the group's process, the treatment of uncertainty is, however, a well-known problem in other disciplines and so in this section we will review those papers which focus on the combination of probabilistic information from a purely statistical approach (see Clemen and Winkler 1999; Genest and Zidek 1986). In general, we might consider these methods as analytical models operating on the individual probability distributions to produce a single "combined" probability distribution. These approaches can generally be further distinguished into axiomatic approaches (by considering a set of assumptions that the combination criteria might satisfy) and Bayesian approaches:

– Axiomatic approach: the following common functions deal with belief aggregation:
  (i) *Linear Opinion Pool* where the group probability, $Pr(G)$, is obtained as the weighted arithmetic average over the individual probabilities, $Pr(V_i), i = 1, \ldots, n$, i.e. $Pr(G) = \sum_{i=1}^{n} w_i Pr(V_i)$, with w_i being the weights totaling one.
  (ii) *Logarithmic Opinion Pool* (weighted geometric average) defined as $Pr(G) = \alpha \prod_{i=1}^{n} Pr(V_i)^{w_i}$, with α being a normalization constant and the weights w_i (called expert weights) are typically restricted to total one. If the weights are equal to $1/n$, then the combined distribution is proportional to the geometric average.
– Bayesian Approach (Genest and Zidek 1986; Clemen and Winkler 1999): this has been used to combine expert information by taking into account the so-called Naive Bayes assumption. In our context, in order to obtain efficient combinations, individual opinions are assumed to be conditionally independent given the group vote.

3 Modeling group decision networks

The purpose of this paper is to develop a general methodology based on the Bayesian network (BN) formalism for modeling those uncertainties that appear in both the interactions between group members and the processes leading to the final choice or recommendation. For example, let us imagine that we want to advise a group of tourists to visit a particular monument or not. In such a situation, we should assume that the individuals in the group are unfamiliar with the monument (or item to be recommended). Each group member might speculate about their possible preference for visiting this monument and this is necessarily uncertain. Nevertheless, the group recommendations must be obtained by aggregating these preferences.

Individual preferences can be computed by considering two alternatives: the first considers content information (such as a description of the monument, location, etc.) and the second considers how people with similar tastes rated this monument in the past (for instance, `dislike` or `like`). This is the approach followed in this paper where the similarity between users will be computed by considering how common items have been rated. Following the classification presented in Sect. 2, our GRS can therefore be categorized as CF+AR+PM.

As a collaborative approach, our model will inherit most of the disadvantages of classical collaborative filtering approaches. For example, the system cannot draw any inferences about items for which it has not yet gathered sufficient information, i.e. we also have the First-Rater problem. Similarly, we also inherit the Cold-Start problem since it is difficult to recommend items to new users who have not submitted any ratings. Without any information about the user, the system is unable to guess user preferences and generate recommendations until a few items have been rated.

For our information sources, we will consider a database of ratings **R** (which is usually extremely sparse) to store user ratings for the observed items. For example, Table 2 shows the ratings given by each user U_i for an item I_j using the values 1 = `dislike` and 2 = `like` (the value $'-'$ represents the fact that the user has not seen the item).

Table 2 Database of user ratings

	U_0	U_1	U_2	U_3	U_4	U_5
I_1	1	1	2	2	1	1
I_2	1	–	2	–	2	2
I_3	1	1	2	1	1	2
I_4	2	–	1	–	1	2
I_5	2	2	1	1	1	1
I_6	2	2	–	2	2	2
I_7	2	–	–	–	1	2

In order to achieve this objective, our aim is to build a BN where two components might be considered. The first, described in Sect. 3.1 relates to the collaborative component of the recommender system. Both the topology of this component and the probability values will be learned from a set of past user ratings, and this will be used to compute a probability distribution representing the preferences of each group member for a given item. The second component will be used to merge these preferences in order to reach the final group opinion. This component is modeled using a BN with a fixed structure given the group members, and the weights will be computed based on the ratings provided by the group members (see Sect. 3.2).

3.1 BN-based collaborative component

In this section, we will briefly describe this component (those readers interested in further details can consult (de Campos et al. 2008)). Our objective is to model how each user should rate an item. In order to represent relationships between users, we shall include a node, U_i, for each user in the system. We use $\mathcal{U}$ to denote the set of user nodes, i.e. $\mathcal{U} = \{U_1, \ldots, U_n\}$. The user variable U_i will therefore represent the probability distribution associated to its rating pattern. For instance, using the data in Table 2, each node will store two probability values representing the probability of U_i liking ($Pr(U_i = 2)$) or disliking ($Pr(U_i = 1)$) an item.

In order to facilitate the presence of dependence relationships between individuals in the model (to avoid a possibly complex network topology), we propose that a new set of nodes $\mathcal{V}$ be included to denote collaborative ratings. There is one collaborative node for each user in the system, i.e. $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$. These nodes will represent a probability distribution over ratings, and they will therefore take their values in the same domain as $\mathcal{U}$.

3.1.1 Learning stage

Given an active user, the parent set of the variable V_a in the graph, $Pa(V_a)$, will be learnt from the database of votes, **R**. This set will contain those user variables, $U_b \in \mathcal{U}$, where U_a and U_b are most similar in taste, i.e. the best neighbors for the active user. Given a similarity measure, the set $Pa(V_a)$ can therefore be obtained by using a threshold or by only considering the first p variables in the ranking (see Fig. 2). It should be

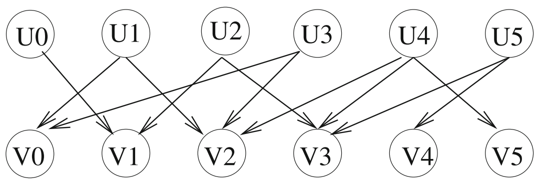


Fig. 2 Collaborative Recommending System Topology

noted that we do not include the links between $U_i \longrightarrow V_i, \forall i$, since we are modeling a collaborative rating scheme where (assuming that the item being recommended has not been observed by the active user) the predicted rating will only depend on those ratings given by its neighbors.

The similarity measure proposed in this paper is a combination of two different, but complementary, criteria: vote correlation between common items and the overlap degree, i.e.

$$sim(U_a, U_b) = abs(PCC(U_a, U_b)) \times D(U_a, U_b) \tag{1}$$

The first criterion, which is normally used as the basis for calculating the weights in different collaborative systems, attempts to capture those similar users, i.e. those with the highest absolute value of Pearson's correlation coefficient defined as

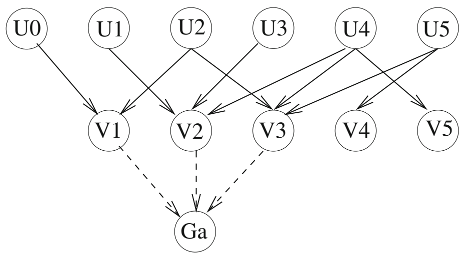
$$PCC(U_a, U_b) = \frac{\sum_j (r_{a,j} - \overline{r}_a)(r_{b,j} - \overline{r}_b)}{\sqrt{\sum_j (r_{a,j} - \overline{r}_a)^2 \sum_j (r_{b,j} - \overline{r}_b)^2}} \tag{2}$$

where the summations over j are over those items for which users U_a and U_b have recorded votes and $\overline{r}_a$ is the mean vote for user U_a. It should be noted that PCC ranges from $+1$ to -1: $+1$ means that there is a perfect positive linear relationship between users; -1 means that there is a perfect negative linear relationship; a correlation of 0 means that there is no relationship. Therefore, when there are no common items in U_a and U_b voting records, then $PCC(U_a, U_b) = 0$ by default. In our approach, by using the absolute value of PCC, $abs(PCC)$, we consider that both positively (those with similar ratings) and negatively correlated users (those with opposite tastes) might help[2] to predict an active user's final rating.

The second criterion tries to penalize those highly correlated neighbors which are based on very few co-rated items, which have proved to be bad predictors (Herlocker et al. 1999). We might therefore take into account the number of items that both U_a and U_b rated simultaneously, i.e. their overlap degree. In particular, we consider that the quality of U_b as the parent of variable U_a is directly related with the probability of a user U_a rating an item which has been also rated by U_b. This criterion can be defined by the following expression:

[2] For instance, if whenever U_b rates as `like` U_a rates with `dislike`, then knowing that U_b had rated an item with `like` provides information about U_a's possible rating.

Fig. 3 Modeling groups



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

where $I(U)$ is the set of items rated by user U in the data set. It should be noted that we are not considering the particular votes, merely whether the users rated an item or not.

3.2 Modeling the group component

As mentioned previously, since groups are usually created by their members, we shall not consider how groups are formed or how they are managed. We shall 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 predict ratings for groups.

We propose to identify a group G as a new node in the BN. Since the recommendations are made by considering the preferences of its members, we propose that the parents ($Pa(G)$) of the group node (G) will be the set of nodes in $\mathcal{V}$ representing its individuals. In this case, we are modeling that the predictions of the group's ratings will depend on the collaborative predictions obtained for each of its members. Figure 3 illustrates a group G_a with three members: V_1, V_2, and V_3. We use dashed lines to represent user-group relations since we assume that the composition of the group is known.

In this paper, we will focus on how different aggregation strategies can be represented in our BN-based model. In order to maintain generality (so that the proposed aggregation mechanisms can be applied in more general situations[3]), we will use the following independence assumption: *given that we know the opinion (ratings) of all the group members, group opinion does not change (it is independent) if the state of any other variable in system* X_i *is known, i.e.* $I(G, X_i|Pa(G)), \forall X_i \notin Pa(G_i)$. It is important to remember that in certain domains this restriction might be very restrictive. For example, it might also be possible to consider other factors that would affect the group rating such as the context. Nevertheless, the study of how to include these factors in the model is beyond the scope of this paper.

[3] For example, it might be used to combine multiple classifiers (Kittler et al. 1998; Abellán and Masegosa 2007) where the new cases will be classified by considering all the results obtained by each classifier.

Table 3 Stored probability values

$P(U_0)$, $P(U_1)$, $P(U_2)$, $P(U_3)$, $P(U_4)$, $P(U_5)$				
$P(V_1 \mid U_0, U_2)$	$P(v_{1,1} \mid 1, 1)$	$P(v_{1,1} \mid 1, 2)$	$P(v_{1,1} \mid 2, 1)$	$P(v_{1,1} \mid 2, 2)$
$P(V_2 \mid U_1, U_3, U_4)$	$P(V_2 \mid 1, 1, 1)$	$P(V_2 \mid 1, 1, 2)$	$P(V_2 \mid 1, 2, 1)$	$P(V_2 \mid 1, 2, 2)$
	$P(v_{2,1} \mid 2, 1, 1)$	$P(v_{2,1} \mid 2, 1, 2)$	$P(v_{2,1} \mid 2, 2, 1)$	$P(v_{2,1} \mid 2, 2, 2)$
$P(V_3 \mid U_2, U_4, U_5)$	$P(v_{3,1} \mid 1, 1, 1)$	$P(v_{3,1} \mid 1, 1, 2)$	$P(v_{3,1} \mid 1, 2, 1)$	$P(v_{3,1} \mid 1, 2, 2)$
	$P(v_{3,1} \mid 2, 1, 1)$	$P(v_{3,1} \mid 2, 1, 2)$	$P(v_{3,1} \mid 2, 2, 1)$	$P(v_{3,1} \mid 2, 2, 2)$
$P(G_a \mid V_1, V_2, V_3)$	$P(g_{a,1} \mid 1, 1, 1)$	$P(g_{a,1} \mid 1, 1, 2)$	$P(g_{a,1} \mid 1, 2, 1)$	$P(g_{a,1} \mid 1, 2, 2)$
	$P(g_{a,1} \mid 2, 1, 1)$	$P(g_{a,1} \mid 2, 1, 2)$	$P(g_{a,1} \mid 2, 2, 1)$	$P(g_{a,1} \mid 2, 2, 2)$

In order to complete the BN-based model, it is necessary to estimate the local probabilities that must be stored in the nodes. In particular, each node X_i has a set of conditional probability distributions, $P(x_i \mid pa(X_i))$ (except root nodes that store marginal probability distributions).[4] For each possible configuration $pa(X_i)$ of the parent set $Pa(X_i)$, these distributions quantify the effect that the parents have on the node X_i. In our case, these probabilities are used to encode both the strength of the user-user interactions and the processes leading to the final choice or recommendation for the group. In Table 3, we show those probability distributions stored in the example of Fig. 3, where for instance $P(V_2 \mid 1, 1, 2)$ represents $P(V_2 \mid u_{1,1}, u_{3,1}, u_{4,2})$. The method of assessing the particular values will be discussed in Sect. 4.

3.3 How to predict the group rating: inference

Once the BN is completed, it can be used to perform inference tasks. In our case, we are interested in the prediction of the group's rating for an unobserved item, I. As evidence, we will consider how this product was rated in the past.[5] The problem therefore comes down to computing the conditional (posterior) probability distribution for the target group G_a given the evidence, i.e. $Pr(G_a \mid ev)$. For instance, let us assume that we want to predict the rating given by G_a in Fig. 3 to item I_7. If we look at Table 2, the evidences are $ev = \{U_0 = 2, U_4 = 1, U_5 = 2\}$ and the problem is to compute $Pr(G_a = 1 \mid u_{0,2}, u_{4,1}, u_{5,2})$.

Since the BN is a concise representation of a joint distribution, we could propagate the observed evidence through the network towards group variables. This propagation implies a marginalization process (summing out over uninstantiated variables).

[4] Throughout this paper we will use upper-case letters to denote variables and lower-case letters to denote the particular instantiation. More specifically, we use v_i to denote a general value of variable V_i and $v_{i,j}$ to indicate that V_i takes the jth-value.

[5] It should be noted that we consider that no member of the group has observed the items beforehand and therefore the evidences are over the values taken by variables in $\mathcal{U}$. In the case of a group member (let us say U_i) having also previously rated I, we shall instantiate both node U_i and V_i to the value of the given ratings. The instantiation of V_i will imply that there is no uncertainty about its rating when the information is combined at a group level. Nevertheless, the computations are more complex in this situation.

A general scheme might be:

$$Pr(G_a = s|ev) = \sum_{\mathcal{V}} Pr(G_a = s|v_1, \ldots, v_k) Pr(v_1, \ldots, v_k|ev) \tag{3}$$

where $v_1, \ldots, v_k$ represents a given configuration of the collaborative variables (parent set) $Pa(G_a)$ and the sum is over the exponential number of possible configurations, and this requires an exponential time $O(r^{|Pa(G_a)|})$ with r being the number of candidate ratings. Considering that the evidence belongs to $\mathcal{U}$, the joint probability over the collaborative variables might be computed as

$$Pr(v_1, \ldots, v_k|ev) = \sum_{\mathcal{U}^-} \prod_{i=1}^{k} Pr(V_i = v_i|u^-, ev) Pr(u^-) \tag{4}$$

where the sum is over all the possible configurations, u^-, of the set of uninstantiated user variables, denoted by $\mathcal{U}^-$, also requiring an exponential time, $O(r^{|\mathcal{U}^-|})$.

These computations should be performed for each group variable when there is an item to be recommended. As there is usually a large set of groups in the system, this process becomes computationally expensive and reducing computational complexity becomes a key design parameter, especially if the objective is to obtain scalable strategies which can be implemented in real-time applications.

In order to tackle this problem, we propose the use of canonical models to represent conditional probability distributions. By means of these models, we can reduce the number of probability values stored and develop specific inference algorithms. In those cases where the computation of $Pr(V_1, \ldots, V_k|ev)$ is complicated, we also propose to approximate these values by using extra independence assumptions (see Sect. 5).

4 Estimating the strength of the users' interactions

In terms of assessing the probability values, we must distinguish between roots in the graph, nodes in $\mathcal{U}$, and the remaining nodes. In particular, for every user node U_k, we need to assess the prior probability distribution over the user's rating pattern, i.e. the probability of user U_k rating with a given value s, $1 \leq s \leq r$. For example, considering the relative frequency and the data in Table 2, we will obtain $Pr(U_3 = 1) = 2/4 = 0.5$ and $Pr(U_5 = 1) = 2/7 = 0.286$.

For each non-root variable, we must store an exponential number of conditional probability distributions: one probability distribution for each possible configuration of its parent set. The assessment, storage, and manipulation of these probability values can be quite complex, especially if we consider that the number of similar users (parents of the nodes in $\mathcal{V}$) and the size of the groups might be large when real group recommending applications are considered. We therefore propose the use of different canonical models to represent these conditional probabilities. By using this representation, it might be possible to reduce the problem of data sparsity (it is quite probable that many configurations lack data), leading to important savings in storage (we only need to store a linear number of probability values) and more efficient inference algorithms (see Sect. 5).

4.1 Probabilities of the collaborative component

The probabilities in the collaborative component (nodes in $\mathcal{V}$) might be estimated from the data set of past user ratings. For a given node V_i, we must define the conditional probability distribution $Pr(v_{i,j}|pa(V_i))$ for each configuration $pa(V_i)$. We propose the use of the following canonical model (studied in detail in de Campos et al. 2008) where an additive behavior of the collaborative nodes is assumed, thereby enabling the data sparsity problem to be tackled:

Definition 1 (*Canonical weighted sum)* Let X_i be a node in a BN, let $Pa(X_i)$ be the parent set of X_i, and let Y_k be the kth parent of X_i in the BN. By using a canonical weighted sum, the set of conditional probability distributions stored at node X_i are then represented by means of

$$Pr(x_{i,j}|pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \tag{5}$$

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 (effects) measuring how this lth value of variable Y_k describes the jth state of node X_i. The only restriction that we must impose is that the weights are a set of non-negative values verifying that

$$\sum_{j=1}^{r} \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) = 1, \quad \forall\ pa(X_i)$$

It is interesting to note that by defining how to compute the weights $w(y_{k,l}, x_{i,j})$, we can control individual bias[6] and the relative quality (importance) of the parents for the predicting variable, X_i.

The problem now is how to estimate those weights given by similar users, i.e. $U_b \in Pa(V_a)$. Following (de Campos et al. 2008), we consider that $w(u_{b,t}, v_{a,s})$ (i.e. the effect of user U_b rating with value t when it comes to predicting the rating of V_a) can be computed by means of

$$w(u_{b,t}, v_{a,s}) = w_{b,a} \frac{N^*(u_{b,t}, u_{a,s}) + 1/r}{N^*(u_{b,t}) + 1}, \quad 1 \le t, \quad s \le r. \tag{6}$$

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

[6] Bias refers to a user's preference for a particular vote (some users tend to rate with high values whereas others prefer to use lower ones) and ability to predict X_i judgments.

U_a, and $N^*(u_{b,t})$ is the number of items in $I(U_a) \cap I(U_b)$ rated with value t by user U_b. In this expression, $w_{b,a}$ represents the relative importance of the parent. In this paper, we assume that all the users are equally important, i.e. $w_{b,a} = 1/|Pa(V_a)|$.

In our example, and focusing on node V_2, we must estimate 2^3 conditional probability distributions. Using Eq. 5, the probability $Pr(v_{2,2}|u_{1,1}, u_{3,1}, u_{4,2})$ is equal to $w(u_{1,1}, v_{2,2}) + w(u_{3,1}, v_{2,2}) + w(u_{4,2}, v_{2,2})$. Using the data in Table 2, these weights are $w(u_{1,1}, v_{2,2}) = 0.278$, $w(u_{3,1}, v_{2,2}) = 0.167$ and $w(u_{4,2}, v_{2,2}) = 0.25$ and therefore $Pr(v_{2,2}|1, 1, 2) = 0.695$.

4.2 Modeling social value functions

The objective of this section is to consider how conditional probability distributions for group nodes can be estimated. These distributions might be considered a "social value function", describing how member opinions affect the group's recommendation. For instance, let G_i be a group with six individuals rating with 1, 5, 5, 4, 5 and 4, respectively, using a rating domain from 1 to 5. In this case, the related configuration is $pa(G_i) = (v_{1,1}, v_{2,5}, v_{3,5}, v_{4,4}, v_{5,5}, v_{6,4})$.

Since we must assess the probability of G_i voting with a value k for each possible configuration $pa(G_i)$ (i.e. $P(G_i|pa(G_i))$) and taking into account that the size of the group might be large, we again propose the use of canonical models. By means of these models, the probability values needed will be computed as a deterministic function of the particular values of the configuration, thereby entailing an important saving in storage.

Definition 2 (*Canonical gate*) A group node G_i is said to represent a canonical combination criterion if given a configuration of its parents $pa(G_i)$ the conditional probability distributions can be defined as

$$P(G_i = k|pa(G_i)) = f(k, pa(G_i))$$

Following the ideas in (O'Connor et al. 2001; Masthoff 2004), in this paper we will consider four alternatives:

4.2.1 MAX and MIN gates

In terms of (O'Connor et al. 2001), Maximum (Minimum) gates can be used to model those situations where the group vote is equal to the vote of the most satisfied (or least satisfied, respectively) group member. Thus, considering the example configuration, the group rating is equal to 5 and 1 for the MAX and MIN gate, respectively. Although these gates correspond to extreme situations, it is quite common for small groups to take into account these criteria when making decisions (Masthoff 2004). More formally, these gates can be defined as

$$f(k, pa(G_i)) = \begin{cases} 1 & \text{if } k = \Phi(pa(G_i)) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Table 4 Using canonical models to define the conditional probabilities for the node G_a in Fig. 3

$pa(G)$	AVG	MAJ	MAX	MIN
$v_{1,1}, v_{2,1}, v_{3,1}$	(1,0)	(1,0)	(1,0)	(1,0)
$v_{1,1}, v_{2,1}, v_{3,2}$	(0.67,0.33)	(1,0)	(0,1)	(1,0)
$v_{1,1}, v_{2,2}, v_{3,1}$	(0.67,0.33)	(1,0)	(0,1)	(1,0)
$v_{1,1}, v_{2,2}, v_{3,2}$	(0,33,0.67)	(0,1)	(0,1)	(1,0)
$v_{1,2}, v_{2,1}, v_{3,1}$	(0.67,0.33)	(1,0)	(0,1)	(1,0)
$v_{1,2}, v_{2,1}, v_{3,2}$	(0.33,0.67)	(0,1)	(0,1)	(1,0)
$v_{1,2}, v_{2,2}, v_{3,1}$	(0.33,0.67)	(0,1)	(0,1)	(1,0)
$v_{1,2}, v_{2,2}, v_{3,2}$	(0,1)	(0,1)	(0,1)	(0,1)

The pairs (x_1, x_2) represent the probabilities $Pr(g_{a,1}|pa(G))$ and $Pr(g_{a,2}|pa(G))$, respectively

where $\Phi(pa(G_i))$ is the $\max\{pa(G_i)\}$ and $\min\{pa(G_i)\}$ for the MAX and MIN gates, respectively. For example, Table 4 shows the probability distribution obtained using these canonical gates for the node G_a in Fig. 3.

4.2.2 MAJority gates

Our objective in this section is to model the Majority criterion where the final decision depends on a simple counting of the votes received for each rating from the individuals. The rating which receives the largest number of votes is then selected as the consensus (majority) decision, i.e.

$$f(k, pa(G_i)) = \begin{cases} \frac{1}{m} & \text{if } k = \arg\max_s count(s, pa(G_i)) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $count(s, pa(G_i))$ is a function returning the number of occurrences of the state s in the configuration $pa(G_i)$, and m is the number of states where $count(s, pa(G_i))$ reaches the maximum value. It should be noted that we are assuming that all members have the same power (i.e. one-person-one-vote). In the previous configuration, the group rating will be 5 because this was the rating given by 3 out of 6 individuals. See Table 4 for an example.

4.2.3 AVeraGe gates

Our objective with this gate is to model those situations where the group rating can be considered as the average of individual ratings. This criterion can be modeled in a similar way to before, i.e.

$$f(k, pa(G_i)) = \begin{cases} 1 & \text{if } k = AVG(pa(G_i)) \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

where $AVG(pa(G_i)) = round\left(\frac{1}{|Pa(G_i)|}\sum_j v_j\right)$, and where v_j is the rating of the jth-parent of G_i. Thus, in the above example, the group rating is defined as $AVG(1, 5, 5, 4, 5, 4) = 4$.

We should mention that although this might be the formal definition of the average gate, in this paper we use a canonical weighted sum-based representation which eventually attempts to recommend the same rating to the group but using much more efficient inference mechanisms. In particular, the weights $w(v_{j,k}, g_{i,s})$ are defined as $1/|Pa(G_i)|$ if $k = s$ and 0 otherwise. An example is shown in the first column in Table 4.

One important fact is that under this representation, the predicted rating must be the group's posterior expected (mean) value which is defined as the sum of the posterior probability of each rating multiplied by the rating value, i.e. $\sum_{s=1}^{r} s \times Pr(G_i = s|ev)$ (see Appendix A for more details). The predicted rating for the example configuration is therefore computed as $1 \times 1/6 + 2 \times 0 + 3 \times 0 + 4 \times 2/6 + 5 \times 3/6 = 4$. At this point, we should mention that whenever we talk about the AVG gate in this paper we are considering this representation.

5 Inference with canonical models

In this section, we will present the specially designed propagation algorithms to ensure efficient computations. Since we consider past user ratings as evidence, a top-down propagation mechanism can be designed, starting with those nodes at the user layer where $Pr(U_i|ev)$ is computed. These probabilities are then used to compute the posterior probabilities at the $\mathcal{V}$ layer (see Eq. 4) which are eventually used to compute the posterior probabilities for the group layer, i.e $Pr(G_a|ev)$ (see Eq. 3). Our GRS therefore follows the typical performance of a collaborative RS since these probability values depend on how similar users rated the item.

We will distinguish between the different canonical models used. When using canonical weighted sums, we can compute exact probability values in polynomial time (see Sect. 5.1). When aggregating individual preferences by means of any other canonical model, it is necessary to use the independence assumptions below in order to reduce the computations needed (see Sects. 5.2 and 5.3).

Independence Assumption: The collaborative ratings are independent given the evidence, i.e.

$$Pr(V_1, \ldots, V_k|ev) = \prod_i Pr(V_i|ev). \tag{10}$$

In view of this assumption, and considering that we use a canonical weighted sum model at the nodes in $\mathcal{V}$, the joint probabilities in Eq. 4 are computed in linear time. Although this assumption might be very restrictive, in our experimentation (see Sect. 7) it has proved to be fruitful and has also been used successfully when combining information for other practical purposes (Clemen and Winkler 1999; Kittler et al. 1998). This performance leads us to believe that it does not matter how accurate the estimates of the posterior probabilities are as long as they help to predict the correct ratings.

5.1 Propagating with the canonical weighted sum

When the canonical weighted sum is used to represent the interaction between variables, the posterior probabilities can be obtained simply by applying the following theorem (see de Campos et al. 2008) which explains how to compute the exact probability values:

Theorem 1 *Let X_a be a node in a BN network, let m_{X_a} be the number of parents of X_a, Y_j be a node in $Pa(X_a)$, and l_{Y_j} the number of states taken by Y_j. If the conditional probability distributions can be expressed under the conditions given by Eq. 5 and the evidence is only on the ancestors of X_a, then the exact posterior probabilities can be computed using the following formula*:

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

It should be noted that propagation can be performed in linear time with the number of parents. We consider this theorem to be important because it expresses the fact that exact propagation can be performed without imposing any independence restriction between the parents of variable X_a (see de Campos et al. 2008). Because the evidences are in nodes in $\mathcal{U}$ in our recommender system, we will therefore obtain the exact posterior probabilities in all the nodes where the conditional probabilities are represented by means of a canonical weighted sum, as will be the case of the nodes in $\mathcal{V}$ and also when modeling the average criterion in the group nodes.

For example, when we want to predict the rating given by G_a in Fig. 3 to item I_7, and considering the past ratings in Table 2, we must compute $Pr(G_a = 1| u_{0,2}, u_{4,1}, u_{5,2})$. In this situation, the exact posterior probabilities at the $\mathcal{V}$ layer are: $Pr(v_{1,1}|ev) = w(u_{0,2}, v_{1,1}) + Pr(u_{2,1}) \cdot w(u_{2,1}, v_{1,1}) + Pr(u_{2,2}) \cdot w(u_{2,2}, v_{1,1}) = 0.277$; $Pr(v_{2,1}|ev) = 0.341$ and $Pr(v_{3,1}|ev) = 0.36$. Similarly, if the group uses an AVG criterion to combine information (represented by a weighted sum gate), the exact posterior values at group nodes are $Pr(g_{a,1}|ev) = 0.326$ and $Pr(g_{a,2}|ev) = 0.674$ and the predicted rating is $round\ (1 \times 0.326 + 2 \times 0.674) = 2$.

5.2 Propagating with majority gates

One key idea behind the majority criterion is that the order in which the individuals are considered does not matter, and therefore there are many different configurations collapsing to the same situation. For example, let us consider that four individuals vote with 1 and one individual votes with 2. In this case, there are five different configurations representing the same situation, i.e. $pa_1(G_i) = \{2, 1, 1, 1, 1\}$, $pa_2(G_i) = \{1, 2, 1, 1, 1\}$, $pa_3(G_i) = \{1, 1, 2, 1, 1\}$, $pa_4(G_i) = \{1, 1, 1, 2, 1\}$ and $pa_5(G_i) = \{1, 1, 1, 1, 2\}$. It should be noted that since order is not a factor, we might talk about combinations with repetition,[7] denoted by δ. Therefore, the above configurations should be represented by $\delta(G_i) =< 1, 1, 1, 1, 2 >$.

In this situation, all the probabilities $Pr(G_i = s|pa_j(G_i))$, such that pa_j can be matched to the same combination δ, have the same values (in our case, $Pr(G_i = 1|pa_j(G_i)) = 1, \forall 1 \leq j \leq 5$). This can be exploited in order to efficiently perform the propagation processes in Eq. 3. In particular, the following theorem shows that we need only take into account those probabilities associated with combinations with repetition in order to propagate individual rating probabilities:

Theorem 2 *Let G_i be a group node in a BN whose conditional probability distributions are represented using a majority gate, let $\Delta(G_i)$ be the set of possible combinations repeating the values in its parent set, $Pa(G_i)$, then*

$$Pr(G_i = s|ev) = \sum_{\delta(G_i)\in\Delta(G_i)} Pr(G_i = s|\delta(G_i))Pr(\delta(G_i)|ev) \qquad (11)$$

The proof of this theorem can be found in Appendix B. Although this theorem reduces the number of necessary computations in Eq. 3, an exponential number of computations will be needed in order to obtain the joint probabilities $Pr(\delta(G_i)|ev)$.

In order to ensure the scalability of the approach, we will approximate these values by considering that collaborative ratings are independent given the evidence (Eq. 10). In Appendix B, we show how to compute the required probabilities in a running time on the order of $O(rn^r)$. Taking into account that in many situations $r << n$, this entails important savings with respect to the $O(r^n)$ needed by the classical approach.

Following on with the previous example, we will have four possible combinations for V_1, $V2$ and V_3 where (assuming independence) the posterior probabilities are $Pr(< 1, 1, 1 > |ev) = 0.034$, $Pr(< 1, 1, 2 > |ev) = 0.215$, $Pr(< 1, 2, 2 > |ev) = 0.446$ and $Pr(< 2, 2, 2 > |ev) = 0.305$. Then, following a majority strategy, $Pr(G_a = 1|ev) = 0.034 + 0.215 = 0.249$ and $Pr(G_a = 2|ev) = 0.751$. Once again, the recommended rating is 2 as this is the most likely posterior probability.

5.3 Propagation with MIN and MAX gates

When propagating with MAX and MIN gates, we will also assume that a posterior probability for the collaborative nodes is independent given the evidence. 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. It can be proved that under the above independence assumption (Eq. 10), the probability distribution $Pr(G_a|ev)$ can be computed easily and efficiently (in a linear order to the number of group members).

Min-Gate Assume $1 < 2 < \cdots < r$

- $Pr(G_a = r|ev) = \prod_{i=1}^{m} Pr(V_i = r|ev) = \prod_{i=1}^{m} Pr(v_{i,r}|ev)$

[7] Since the number of parents in G_i is n and each parent has r different states, we find that the number of possible combinations with repetition is $CR_n^r = (n + r - 1)!/(n!(r - 1)!)$.

– $Pr(G_a = k|ev)$, for $k = 1, \ldots, r-1$, is equal to

$$\left(\prod_{i=1}^{m} Pr(V_i \geq k|ev)\right) - Pr(G_a > k|ev).$$

where $Pr(X \geq k) = \sum_{j=k}^{r} Pr(X = j)$.

Max-Gate Assume $1 < 2 < \cdots < r$

– $Pr(G_a = 1|ev) = \prod_{i=1}^{m} Pr(V_i = 1|ev) = \prod_{i=1}^{m} Pr(v_{i,1}|ev)$

– $Pr(G_a = k|ev)$, for $k = 2, \ldots, r$, is equal to

$$\left(\prod_{i=1}^{m} Pr(V_i \leq k|ev)\right) - Pr(G_a < k|ev)$$

where $Pr(X \leq k) = \sum_{j=1}^{k} Pr(X = j)$.

If we consider our example, the posterior probabilities for the MIN gate are $Pr(G_a = 2|ev) = \prod_{i=1}^{3} Pr(v_{i,2}|ev) = 0.305$; $Pr(G_a = 1|ev) = 0.695$ and the recommended rating is 1, whereas if we consider the MAX gate $Pr(G_a = 1|ev) = \prod_{i=1}^{3} Pr(v_{i,1}|ev) = 0.034$; $Pr(G_a = 2|ev) = 0.966$ and the decision is to recommend the rating 2.

6 Experimental framework

There are various reasons why GRSs are difficult to evaluate. The first is that the evaluation criterion differs according to the goals for which the RS has been developed. We can find situations where the criteria used to measure system performance are user satisfaction or participation (O'Connor et al. 2001; Masthoff 2004; McCarthy and Anagnost 2000) whereas in other GRSs the objective is to explore the ability of the system to merge user profiles or reach a consensus (Jameson 2004; Yu et al. 2006).

The second reason why evaluation is difficult is the absence of public data sets for performing the evaluation. Most work into GRS evaluation has focused on live user experiments. In this case, we can distinguish between the work using controlled groups that have been (directly or indirectly) asked about the aggregation strategies they might use (Masthoff 2004; Masthoff and Gatt 2006) and those systems, such as Polylens (O'Connor et al. 2001) and MusicFX (McCarthy and Anagnost 2000), which have made the system available to a community of users. In this case, field studies were performed to evaluate system performance. The combination of these two factors makes it extremely difficult to compare the performance between different GRSs.

This situation clearly differs from that of recommending for a single user. In this context, the evaluation of system accuracy, i.e. the system's ability to predict individual ratings, has become a standard approach (Herlocker et al. 2004). Surprisingly, little attention has been paid in GRS literature to evaluating system accuracy. We believe that one of the main reasons for this is that it is difficult to access real group ratings since in many cases group composition is ephemeral, and, to the best of our

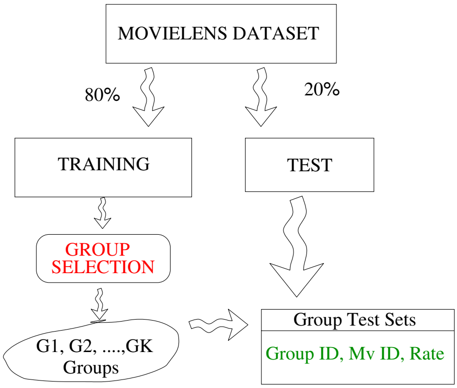


Fig. 4 Building the group data sets

knowledge, no data sets exist with this kind of information. On the other hand, we can consider that in real scenarios each group might rate using different aggregation strategies. For instance, some groups might use a *least misery strategy* whereas others might use an *average strategy*. In these situations, a blind algorithm, which does not take into account how the group rates an item, may be inappropriate for predicting group ratings.

Our goal in this experimentation is to discover how the use of the different uncertainties emerging in a group recommending process might affect system performance. We believe the best alternative for measuring system performance is to take into account accuracy criteria in an automatic evaluation over a semi-simulated data set (see below). We believe that by means of this evaluation, we can conduct large experiments that should validate our conclusions.

6.1 The data sets

We decided to use the MovieLens[8] data set. With the idea of using 5-fold cross validation, we have used 5 different data subsets, each obtained by splitting MovieLens into two disjoint sets, the first for training (with 80% of the data) and the second for testing (with 20% of the data).

Training data has been used for two different purposes. Firstly, we have used this data to learn the collaborative component of the system. Following (Herlocker et al. 1999), we have considered a fixed number of 10 parents (similar users) for each node in $\mathcal{V}$ (see Sect. 3.1). Using this component independently, we might compute for each member of a group, V_k, a probability distribution representing how this individual might rate an unseen movie, i.e. $Pr(V_k = s|ev)$. Training data has also been used to determine group composition, i.e. which individuals form a group (see Fig. 4).

[8] MovieLens was collected by the GroupLens Research Project at the University of Minnesota. The data set contains 1682 movies and 943 users, containing 100,000 transactions where a user rates a movie using a scale from 1 to 5.

We have used two different criteria in an attempt to capture different processes behind the creation of a group:

C1 Implementing the idea of *the group of my buddies*, we set each user as the group administrator and look for similar users (those who are positively correlated with the administrator in the training data set). We then select those groups for a fixed number of individuals (let us say n) with the only restriction being that they have at least rated (seen) one common movie (considering the experimental data, the members of the groups have rated a mean of 13.63 common movies). It should be noted that since similarities are not transitive, this criterion does not necessarily imply that groups have highly correlated members.

C2 Secondly, we have decided to fix a group (also with n individuals) with the only restriction being that all group members must rate at least four common movies in the training sets, independently of the given ratings. This alternative can be used to represent *circumstantial groups* such as those obtained by randomly selecting people as they leave a cinema. With this criterion, it is plausible that while individuals might watch a movie together, they might have different preferences.

The group test sets are obtained from each of the original MovieLens test sets, thereby ensuring that no data in the test sets has been used for learning purposes. Whenever we find a movie in an original MovieLens test set that has been rated by every group member, we include the tuple (*group ID, movie ID, group-rating*) in the respective group test data set. Since we know the real user ratings for this movie, the "true" group-rating will be obtained by considering the mechanism used by the group to aggregate the individuals ratings. This mechanism might be implemented by means of a deterministic function `CombineRate`$(r_1, \ldots, r_n)$.[9] For example, given a group of five individuals and the ratings $r_1 = 5, r_2 = 5, r_3 = 2, r_4 = 3, r_5 = 4$, then depending on the aggregation function used, the "true" group rating will be: 5 for the *majority* function, 2 for the *minimum* criterion, 4 if we consider an *average* strategy, etc.

By combining the strategy used by a group to rate a movie, i.e. average (AVG), majority (MAJ), maximum (MAX) and minimum (MIN), and the criterion used to form a group (C1 and C2), we therefore obtain 8 different data sets. More specifically, by fixing the group size to 5, we have found a mean of 108 (406) different groups and 115 (1752) group-movie pairs in the test set for C1 criterion (C2 criterion, respectively).

7 Experimentation: predicting a group's rating

The aim of this experimentation is to measure the effect of individual uncertainties on each different aggregation criterion. In order to focus on this aim, we shall assume that we know how the group combines individual ratings, i.e. we know whether the group rating is obtained by means of the *AVG, MAJ, MAX* or *MIN* of a set of individual ratings (what happens when this information is unknown will be discussed in Sect. 8).

[9] A similar mechanism for building the group rating has been used in Chen et al. (2008) to measure the effect of sparsity on the recommendations.

Table 5 Baseline and BN-based group aggregation mechanisms

Baseline	Using group layer
For each $V_k \in G$ do	$Pr(G\|ev) =$ CombineProb($Pr_1, \ldots, Pr_n$)
$r_k =$ RateSelection(Pr_k)	G_rate = RateSelection($Pr(G\|ev)$)
G_rate = CombineRate($r_1, \ldots, r_n$)	

The following method is proposed: for each group-item pair in the group test set, the first step is to instantiate those users who had rated the movie with the given rating (these ratings will be considered as the *evidence*, see Sect. 3.3). This evidence will be then propagated to the collaborative nodes, $\mathcal{V}$, representing the group's members. This information must finally be combined in order to obtain the predicted rating. At this point, it should be remembered that we assume that no member of the group has seen the movie and it is not possible to know how a node V_i might rate the movie. We will consider two different alternatives to combine this uncertain information:

– This alternative, which could be considered the *Baseline* (see left-hand side of Table 5), is a two-step approach: firstly, each group member makes a decision about the score that he or she might use to rate the movie (RateSelection process). These $r_1, \ldots, r_n$ precise ratings are then combined using the particular combination strategy (AVG, MAJ, MAX or MIN) used by the group when making decisions.
– The second method consists in first using canonical models to combine individual probability distributions into a single group distribution. This probability distribution ideally represents the group voting pattern. In the algorithm (see right-hand side of Table 5) this process is denoted as CombineProb. The group probability distribution is then used to select the final group rating (once again using the RateSelection process).

Finally, we will consider how the predicted rating might be selected, i.e. how RateSelection works. In this paper, we will explore two different alternatives:

– The first uses the raw probability values at group nodes (or collaborative nodes for the baseline approach). In particular, the group rating is defined as the posterior average rate, i.e. $rate = \sum_{k=1}^{r} k \times Pr(G = k|ev)$, for the AVG canonical model (see Sect. 4.2.3) and the maximum posterior probability, i.e. rate $= \arg\max_s \{Pr(G = s|ev)\}$, for MAX, MIN and MAJ canonical models.
– The idea behind the second alternative is to only take into account the new piece of evidence that each candidate rating receives. This criterion can be computed by taking into account the difference between the posterior and prior probability values,[10] i.e. $PD(G_a) = Pr(G_a|ev) - Pr(G_a)$. We consider only those ratings where the difference is positive, i.e. the evidence favors these ratings. We have to notice that PD is not a probability measure since $\sum_s PD(G_a = s) < 1$.

[10] It is worth remembering that posterior probabilities are obtained by instantiating the ratings previously given by similar users (the evidence belongs to the nodes in $\mathcal{U}$) and that prior probabilities are obtained by propagating in the network without evidence.

Table 6 Effect of user uncertainty on the combination strategies

Group rate	RS	C1				C2			
		Baseline		BN-Group		Baseline		BN-Group	
		$\%S$	MAE	$\%S$	MAE	$\%S$	MAE	$\%S$	MAE
AVG		62.111	0.387	44.945+	0.596+	63.730	0.369	59.037+	0.420+
	PD	**62.500**	**0.385**	62.060−	0.396−	64.045	0.365	**64.355**−	**0.363**−
MAJ		59.127	0.464	59.543+	0.430−	56.548	0.484	58.621+	0.441+
	PD	59.044	0.464	**59.745**+	**0.428**−	57.447	0.472	**58.926**+	**0.437**+
MAX		77.397	0.226	73.748−	0.285+	68.740	0.319	**75.353**+	**0.263**+
	PD	77.538	0.225	**78.991**+	**0.211**−	69.501	0.311	70.272+	0.300+
MIN		44.964	0.674	13.739+	1.648+	46.987	0.681	29.509+	1.032+
	PD	**46.195**	0.666	42.876+	**0.662**+	46.539	0.684	**49.780**+	**0.578**+

For example, let us consider that the prior probability values (in a rating domain from 1 to 3) are $P(g_{a,1}) = 0.1$, $P(g_{a,2}) = 0.2$ and $P(g_{a,3}) = 0.7$ and let us consider the following posterior values $P(g_{a,1}|ev) = 0.30$, $P(g_{a,2}|ev) = 0.25$ and $P(g_{a,3}|ev) = 0.45$. Thus, if we were to select the maximum posterior rating we would recommend the rating 3 whereas evidence seems to favor the rating 1. Therefore, considering PD, the new measure assigns a mass 0.20, 0.05 and 0 to ratings 1, 2 and 3, respectively. In order to standardize the recommending process, this new measure might be transformed into a probability by means of a proportional normalization process.

The effect of using uncertainty will be measured by considering the accuracy of the recommendations in the two previous situations. We have considered two different metrics (Herlocker et al. 2004): 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.

Table 6 shows the results obtained for the different experiments. The first column represents the criterion used by the group to determine the group rate. Each row represents the results obtained when using this criterion to combine the information with the baseline model and when using the group layer in the BN, denoted by the BN-Group. The rows labeled PD in the second column represent the results obtained when considering the difference between prior and posterior probability distributions. We have highlighted in bold the best alternatives for each particular situation. We use the signs + and − throughout this paper to represent the fact that the results are significantly relevant or irrelevant, respectively, in relation to the baseline model, by using the paired Student's *t*-test (confidence level 0.05).

From this table, we can conclude that when combining uncertain information using BN at a group layer the best option is to use PD to correct the prior bias, particularly in those situations where the minimum and the average gates are used to merge individual ratings. The exception is the use of MAX gate (using the C2 data set) where

Table 7 Using a classical collaborative filtering algorithm

Group rate	C1		C2		Group rate	C1		C2	
	$\%S$	MAE	$\%S$	MAE		$\%S$	MAE	$\%S$	MAE
AVG	49.080+	0.538+	57.986+	0.422+	MAJ	52.713+	0.526+	56.562+	0.478+
MAX	74.655+	0.258+	67.920+	0.328+	MIN	38.255+	0.761+	39.258+	0.760+

the baseline model performs better. In terms of the use of PD in the baseline model, it seems that the results have not been affected significantly. We can also conclude that taking individual uncertainties into account helps to improve the recommendations (we obtain better MAE (%S) values in 7 out of 8 (6 out of 8) experiments). Additionally, the results in Table 6 show that it is possible to order the different aggregation strategies in relation to their accuracy: $MAX \prec AVG \prec MAJ \prec MIN$, where $X \prec Y$ means that X obtains better predictions than Y. We believe that this is due to the bias that MovieLens has towards high rating values (approximately 75% of the ratings have a value which is greater than or equal to 3) and that, in some respects, our collaborative model inherits this bias in the parameters' learning phase. Finally, we can conclude that the way groups are formed is relevant for prediction purposes. Therefore, MIN and AVG canonical models obtain better results when considering circumstantial groups (C2 group set) whereas MAX and MAJ canonical models obtain better results in the case of groups with related individuals (C1 group set).

Finally, and in order to validate our approach, we will compare our predictions with those obtained with a classical collaborative filtering algorithm (Herlocker et al. 1999) under the same conditions.[11] Following our baseline scheme, we first use the classical model to predict how individual users will rate specific items. These predictions are then aggregated by using the corresponding functions. Table 7 presents the obtained results for each execution. From this table, we can conclude that our model is an improvement on those obtained using a classical approach for recommending.

7.1 Effect of the group's size

We will now explore the effect of group size on the recommendations. The experiments were repeated but this time with groups comprising 3, 4, 5 and 6 individuals, except under criterion C1, *groups of my buddies*, where no groups with 6 members could be found. Table 8 presents the MAE metrics obtained after evaluating our models in the same conditions as before using C1 and C2 data sets, respectively. We only show the MAE values obtained using PD for each pair 'XXX#s', where XXX represents the aggregation strategy and #s the group's size, respectively.

[11] In this classical model, the similarity between users is computed using Pearson's correlation measure (Eq. 2). Once similarities are ready, predicting how user U_a will rate an item I_j can be calculated as $rate = \overline{r}_a + \frac{\sum_{h=1}^{m'} sim(U_a,U_h)(r_{h,j}-\overline{r}_h)}{\sum_{h=1}^{m'} |sim(U_a,U_h)|}$ where m' is the number of related users who have also rated item I_j.

Table 8 MAE results for different sized groups

C1 datasets

	AVG3	AVG4	AVG5	AVG6	MAJ3	MAJ4	MAJ5	MAJ6
Base	0.490	0.454	0.385		0.482	0.488	0.464	
BNPD	0.481	0.424	0.396		0.412	0.431	0.428	
	MAX3	MAX4	MAX5	MAX6	MIN3	MIN4	MIN5	MIN6
Base	0.407	0.310	0.225		0.738	0.742	0.666	
BNPD	0.389	0.308	0.211		0.663	0.649	0.662	

C2 datasets

	AVG3	AVG4	AVG5	AVG6	MAJ3	MAJ4	MAJ5	MAJ6
Base	0.469	0.417	0.365	0.327	0.477	0.497	0.472	0.427
BNPD	0.470	0.398	0.363	0.334	0.417	0.450	0.437	0.378
	MAX3	MAX4	MAX5	MAX6	MIN3	MIN4	MIN5	MIN6
Base	0.425	0.362	0.311	0.203	0.724	0.693	0.684	0.716
BNPD	0.398	0.341	0.300	0.199	0.644	0.607	0.578	0.604

It is possible to draw certain conclusions from this data. Firstly, we can say that the relative performance between both the baseline and the proposed BN-based group aggregation model is stable for almost all the experiments (27 out of 30). We can therefore say that it is preferable to combine uncertain information using BN, independently of the size of the group. Nevertheless, group size has an important impact on the quality of the model's predictions. This impact seems to be independent of the criteria used to create the group, C1 or C2, but not on the aggregation strategy used by the group to recommend the final rating. Accordingly, AVG and MAX obtain significantly better predictions (lower MAEs) as the group size increases (this also seems to be the case of MAJ when using C2 data sets) whereas the MIN criterion seems to be more or less stable. For AVG and MAJ combination criteria, this situation might be explained by the fact that as the number of members increases, there is a reduction in the impact of each member on the group prediction. In terms of the performance of extreme rating criteria such as MAX and MIN, this can be explained by considering the bias for rating with higher values in MovieLens, with it being easier to predict high rating values.

8 Uncertainties in the group's rating processes

In the previous section, we assumed that we know how a group combines the information and focused on the effect of uncertainty on user preferences when predicting the group's rating. In this section, we will consider that the process used by the group to rate a given product is also uncertain. In order to tackle this problem we will consider two different situations:

– The first is a situation of total ignorance, i.e. we do not know anything about how the group should combine the information.

Table 9 Predicting without information

Group rate	C1					C2				
	AVG+PD		MAJ+PD		LinOP	AVG+PD		MAJ+PD		LinOP
	Base	BN-G	Base	BN-G	–	Base	BN-G	Base	BN-G	–
AVG	0.385	0.396−	0.486	0.385+	0.386	0.365	0.363−	0.486	0.389+	0.388
MAJ	0.472	0.458−	0.464	0.428−	0.439	0.453	0.440+	0.472	0.437+	0.439
MAX	0.799	0.727+	0.740	0.709+	0.718	0.854	0.808+	0.793	0.788−	0.796
MIN	1.140	1.162+	1.192	1.167+	1.175	1.186	1.220+	1.283	1.243+	1.236
Mean	0.699	0.686	0.721	**0.672**	0.680	0.715	**0.708**	0.759	0.714	0.715
Dev	0.344	0.348	0.338	0.360	0.361	0.380	0.393	0.380	0.395	0.392

– The second considers that although we do not know exactly how a group combines the information, we have a database of previous group ratings. In this case, it might be possible to discover the mechanism used by a group to rate an item from the database of past ratings.

8.1 Total ignorance

Our objective is to study which of these proposed aggregation models is best if we do not know how the group combines the information. This situation, which might be related with the Cold-Start problem, is common when a new group is incorporated into the system and therefore the decision processes are blind. Nevertheless, we have tried to study whether the use of the proposed canonical model might be helpful or not under these circumstances.

Thus, given a new group, it might not be appropriate to determine the group rating by means of extreme canonical models such as MIN, MAX (as confirmed by preliminary experimental results). We will therefore attempt to determine the best option between combining the results using an average or a majority criterion (which in some respects relates to recommending the mean or mode value, respectively). We will also compare our results with those obtained using the classical linear opinion pool (LinOP) where all the users have been considered equivalent for prediction purposes.[12] In this case, the recommended rating is the one that obtains the maximum posterior probability.

In order to study the performance of MAJ and AVG in this situation, we have decided to measure the accuracy of MAJ+PD and AVG+PD when predicting group ratings under the four different combination mechanisms. As in Sect. 7, the experimentation has been conducted by considering the two criteria for forming the groups (groups of buddies and circumstantial users). Table 9 presents the results (we only

[12] It should be noted that this model is equivalent to considering a weighted sum canonical model using an unbiased uniform weighting scheme, i.e. the weights are defined as $w(v_{j,t}, g_{i,s}) = w_j$ if $t = s$, and 0 otherwise). The linear opinion pool can therefore be considered a particular case of the canonical weighted sum gate.

show the MAE metrics[13]). The last two rows of this table contains the mean values and standard deviation for the experiments.

From this table, we can conclude that it is safer to use the canonical models to combine information, and for majority gates, in particular, the MAE values are 6% better for both C1 and C2. Taking into account the uncertainties at an individual level therefore helps to improve the prediction. As before, the model performs differently when consideration is taken of how the groups are formed. On average, all the models obtain better results when groups of buddies (C1) are used than when groups of circumstantial members (C2) are considered. When using C1, the use of MAJ+PD criterion seems preferable to AVG+PD (statistical significance were found) and this might imply that in the case of groups with similar users it is better to use the mode value. When using C2, however, although it was better to use AVG+PD, there is no statistical significance with respect to MAJ+PD. Finally, we would like to mention that we have also studied the performance of the LinOP model when attempting to reduce the effect of the prior probabilities, but in this case no significant differences were obtained (the same mean results were obtained).

8.2 Learning how the groups rate

The aim of this section is to study the use of automatic learning algorithms to represent the group's profiles and their effect on the group's recommendations. There are two main points which must be considered: the first is that we are not imposing any restriction on how a group rates an item, and therefore the learning models might be general; and the second is related to efficiency and scalability since, as we have seen, these are both important in group recommending.

This paper will therefore explore the following two alternatives:

NB The first method uses a Naive Bayes (NB) classifier (Duda and Hart 1973) since it performs well on many data sets, and is simple and efficient. This modeling might be related to the classical Bayesian approach for combining probability distributions. In our context, predicting the group rating can be viewed as a classification problem where the group rating is the class variable and the individual ratings would be the attributes. NB assumes that individual ratings are independent given the group rating and has the advantage that no structural learning is required. In this case, the rating prediction comes down to finding the *rate* such that

$$rate = \arg\max_s P(G = s) \prod_i Pr(v_i | G = s)$$

where v_i is the rating selected for the ith member of the group, V_i, by means of a `RateSelection(`$Pr(V_i|ev)$`)` process as in the previous baseline model (see left-hand side of Table 5).

[13] We have decided not to include the percentage of success in order to reduce the size of the tables. It should be noted that it appears to be correlated with the MAE values, as previously seen in the experiments.

The only parameters that must be estimated from the data sets of past group ratings are the group's prior probability, $P(G)$, and each member's rating conditional probability given that the group has rated with a certain value, $Pr(V_i|G = r)$. These probabilities can easily be estimated from the data sets using Add-$1/r$ smoothing such that

$$Pr(G_a = s) = \frac{N(g_{a,s}) + 1/r}{N(G_a) + 1}$$

where $N(g_{a,s})$ is the number of times that the group rating is s and $N(G_a)$ is the number of times that the group has rated.
For the nodes in $\mathcal{V}$, the conditional probabilities are estimated using

$$Pr(V_i = t|G = s) = \frac{N(v_{i,t}, g_{a,s}) + 1/r}{N(g_{a,s}) + 1}$$

where $N(v_{i,t}, g_{a,s})$ is the number of times that the user rate is t when the group rating is s .

WSG The second alternative consists in using the canonical weighted sum model presented in this paper (see Definition 1) to represent the combination process. In this case, we are assuming that the group rating is determined by aggregating the preferences of the group's members and the only independence restriction that we impose is that since we know how the group's members rate an item, the group rating is independent of the other information sources. As with NB, no structural learning is required in this model.
We must therefore determine each individual's effect on the group rating. In particular, looking at Eq. 5, we must estimate $w(v_{i,k}, g_{a,s})$, i.e. the effect that user V_i rating with the value k has on the group G_a rating with the value s. These weights might be defined as the ratio

$$w(v_{i,k}, g_{a,s}) = \frac{N(v_{i,k}, g_{a,s}) + 1/r}{N(v_{i,k}) + 1}.$$

The main advantages of both models rest on the assumptions used, which reduce the number of parameters to be estimated in several orders of magnitude and also facilitate the inference processes. Both assumptions are, however, the main drawbacks of the models since they are not realistic in this domain.

8.2.1 Experimentation: learning group rating pattern

In this experiment, we will focus on the ability of the proposed models to learn how the groups rate. We have therefore conducted the same experiments as before, but considering the learned group profiles. Table 10 presents the results obtained with both models when considering different combination mechanisms of the group. More specifically, we only show the results obtained using WSG with PD strategy (WSG+PD).

Table 10 Learning parameters from the database of cases

Group rate	C1				C2			
	Naive Bayes		WSG+PD		Naive Bayes		WSG+PD	
	%S	MAE	%S	MAE	%S	MAE	%S	MAE
AVG	55.995	0.574	57.456−	0.442+	57.727	0.471	62.222+	0.387+
MAJ	52.588	0.637	56.037−	0.488+	51.477	0.583	56.180+	0.487+
MAX	72.171	0.449	77.702+	0.244+	67.300	0.407	72.689+	0.284+
MIN	38.738	0.804	42.829+	0.710+	39.955	0.781	45.213+	0.663+
Mean	54.873	0.616	**58.506**	**0.471**	54.115	0.561	**59.076**	**0.455**
Dev	13.736	0.148	14.393	0.191	11.465	0.164	11.486	0.161

In addition, in terms of rate selection for WSG+PD, the average rating over the group probabilities is recommended,[14] i.e. $rate = round\left(\sum_{k=1}^{r} k \times Pr(G = k|ev)\right)$.

From this table, we can conclude that WSG+PD outperforms the Naive Bayes model in all the experiments. Finally, we will compare the results with those obtained in Table 6, which could be considered our goal since in this case we are using the same aggregation strategy as that used by the group. Although we always obtain better results as Table 6 shows, we can conclude that by learning the group rating pattern, we might make predictions which are almost ideal. For instance, the difference between the best MAE values are 0.046, 0.06, 0.033 and 0.048 (0.024, 0.05, 0.021 and 0.086) for AVG, MAJ, MAX and MIN for C1 data sets (C2 data sets, respectively). Although these differences are statistically significant, they give some idea of the ability of the WSG to map different aggregation criteria. This is important because it implies that WSG can be applied safely in those situations where the aggregation criterion used by the group is not known.

9 Conclusions

In this paper, we have proposed a general BN-based model for group recommending and this is an intuitive representation of the relationships between users and groups. The topology of the BN represents those dependence and independence relations considered relevant for modeling the group recommending processes. We should emphasize that only in situations where the required computation is quite complicated have we considered assumptions to reduce computational complexity. Although these assumptions might not be realistic in a given domain, they are necessary if we want to apply the proposed methodology in real applications. The experimental results show the viability of our approach.

With the proposed model we can therefore represent both uncertainty relating to the user's personal views about the relevance of an item, and also uncertainty

[14] The other alternatives were the combination of WSG without PD and the use of the most probable criterion to decide the predicted rating, but worse results were obtained.

relating to the mechanisms used by the group to aggregate individual preferences, mechanisms which are encoded by means of conditional probability distributions stored at the group nodes. In terms of efficiency, these distributions have been assessed by means of canonical models. The use of these models also allows the posterior probabilities to be computed in linear time and this is something which is necessary for deciding the recommended rate. Guidelines have been given for how to estimate the probability values from a data set and how the users interact with the RS. Experimental results demonstrate that by taking uncertainty into account at the individual level when aggregating, better prediction for the groups can be obtained. In addition, the results obtained determine other factors which affect system performance such as how the group is created, the number of individuals in the group, the aggregation function used, etc. It should 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 ratings*) for a single item or for a set of items. Moreover, although this paper is set within the framework of the group recommending problem, the results presented can easily be extended to those disciplines where aggregating information represents an important component, and these disciplines include statistics, decision theory, economics, political science, psychology, etc.

By way of future work, we will attempt to incorporate mechanisms to enable consensus to be reached between group members. In this respect, we can say that selecting the rating r_i with probability 0.55 is not the same as selecting the rating r_i with probability 0.95. It seems clear that the use of these uncertainties will be helpful for reaching consensus in group decisions. We also plan to encode different strategies such as *ensuring some degree of fairness* (Masthoff 2004; Jameson and Smyth 2007) by means of a BN, for instance to ensure that at least 75% of users were satisfied. Another problem worthy of study is to determine in what circumstances it is possible to discover how a group rates a given item, such as for example how to consider the effect of context in the group rating pattern.

Acknowledgements This work has been jointly supported by the Spanish Ministerio de Educación y Ciencia, Junta de Andalucía and the research program Consolider Ingenio 2010, under projects TIN2005-02516, TIC-276 and CSD2007-00018, respectively.

Appendix

This appendix will include any technical results that are not necessary to understand the insights of the model but which are necessary to follow the mainstream of the paper.

A Modeling the average rating with canonical weighted sum gates

The average gate models those situations where the group rating can be considered as the average rating of its members. Although representing this combination criterion using this gate implies important savings in storage, a huge number of computations are required in the inference processes. In this appendix, we will illustrate how the

Table 11 Probability values for average gate and CWS-based criteria for configurations $c_1 = (1, 1, 2, 1, 1, 2)$, $c_2 = (1, 2, 2, 3, 3, 5)$ and $c_3 = (1, 5, 5, 4, 5, 4)$.

	AVG gate: $Pr(G = k\|c_i)$					CWS-based: $Pr(G = k\|c_i)$				
	1	2	3	4	5	1	2	3	4	5
$c1$	1	0	0	0	0	0.666	0.333	0	0	0
$c2$	0	0	1	0	0	0.166	0.333	0.333	0	0.166
$c2$	0	0	0	1	0	0.166	0	0	0.333	0.5

same recommended rating might be computed efficiently by considering a canonical weighted sum-based representation of the average rating in a two-step approach:

– Firstly, for a given configuration $pa(G_i)$, define the probability of the group rating with the value k as the ratio of the number of its members which would rate with k, i.e.

$$Pr(G_i = k|pa(G_i)) = \frac{\sum_{V_j \in Pa(G_i)} R(pa(G_i), V_j, k)}{|Pa(G_i)|}$$

with $R(pa(G_i), V_j, k)$ equal to 1 if user V_j rates with value k in the configuration $pa(G_i)$ and 0 otherwise. This definition is equivalent to aggregating individual ratings by means of a canonical weighted sum approach (see Definition 1) where all users are assumed to be equal for prediction purposes[15] with no bias in the user ratings. These weights might therefore be defined as follows

$$w(v_{j,t}, g_{i,k}) = \begin{cases} \frac{1}{|Pa(G_i)|} & \text{if } k = t, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In Table 11, we present both the probability values for the AVG gate using Eq. 9 and those obtained using Eq. 12 (column labeled with CWS-based) for three example configurations.

– Secondly, determine the group rating as the one obtained by averaging over the group probabilities as

$$rate = round\left\{\sum_{k=1}^{r} k \times Pr(G_i = k|pa(G_i))\right\} \tag{13}$$

Thus, continuing with our example, for configuration c_1 the group rate is $round\{1.333\} = 1$, for configuration c_2 $round\{2.6666\} = 3$, and for configuration c_3 $round\{4\} = 4$, which are the same rating values as those that will be recommended using the average over individual ratings.

[15] It should be noted that in this case we do not consider situations where there are users with high quality opinions (experts). Nevertheless, these could easily be taken into account by adequately modifying the weights.

We should mention that by using a canonical weighted sum representation, we inherit the computational advantages of the canonical weighted sum models in the inference processes (see Sect. 3.3). When we refer to the average canonical model in this paper, denoted by AVG, we are therefore referring to the fact that the group rating is obtained using the conditional probabilities presented in Eq. 12 and that the selected rating is the one obtained using Eq. 13.

B Propagating with majority gates

In this section we will present the computations necessary to understand how the new evidences can be propagated efficiently using majority gates.

Since order is not a factor in the majority criterion, we might speak of combinations with repetition. We will use $\Delta(G_i)$ to denote the set of combinations with repetition from the individual votes in $Pa(G_i)$, and we use $\delta(G_i)$ or $< >$ to denote a single combination. We will say that a configuration pa belongs to the combination δ, denoted by $pa \in \delta$, if the combination δ can be obtained from configuration pa by removing the order constraints.

The following theorem shows that in order to combine the different individual ratings we only need to take into account the probability distributions associated to the set of combinations with repetition:

Theorem 2 *Let G_i be a group node in a BN whose conditional probability distributions are represented using a majority gate, let $\Delta(G_i)$ be the set of possible combinations with repetition of the values in its parent set, $Pa(G_i)$, then*

$$Pr(G_i = s|ev) = \sum_{\delta(G_i)\in\Delta(G_i)} Pr(G_i = s|\delta(G_i))Pr(\delta(G_i)|ev)$$

Proof Considering the independences in the model (see Sect. 3.2) we have that

$$Pr(g_{i,s}|ev) = \sum_{pa(G_i)} Pr(g_{i,s}|pa(G_i))Pr(pa(G_i)|ev)$$

If we consider the set of configurations that can be mapped to the combination $\delta(G_i)$, i.e. $pa(G_i) \in \delta(G_i)$, then

$$Pr(g_{i,s}|ev) = \sum_{\delta(G_i)} \sum_{pa(G_i)\in\delta(G_i)} Pr(g_{i,s}|pa(G_i))Pr(pa(G_i)|ev).$$

Since for the majority gate all configurations mapping to the combination $\delta(G_i)$ have the same conditional probability distribution, $Pr(g_{i,s}|\delta(G_i))$, the right-hand side of the above equality becomes

$$\sum_{\delta(G_i)} Pr(g_{i,s}|\delta(G_i)) \sum_{pa(G_i)\in\delta(G_i)} Pr(pa(G_i)|ev)$$

and finally

$$\sum_{\delta(G_i)} Pr(g_{i,s}|\delta(G_i))Pr(\delta(G_i)|ev). \qquad \square$$

This theorem shows that if we know $Pr(\delta(G_i)|ev)$, the information could be combined with a majority gate in a time which is proportional to the size of CR_n^r, i.e. $O(n^{r-1})$. Taking into account that in many situations $r << n$, this implies important savings in terms of considering the number of possible configurations, $O(r^n)$. For instance, if $n = 20$ and $r = 2$ then $CR_n^r = 21$ whereas the number of configurations (permutations) is more than a million.

B.1 Assuming independence to approximate $Pr(\delta(G_i)|ev)$:

In order to compute $Pr(\delta(G_i)|ev)$, however, we must sum over all the possible configurations in the combination, i.e. $\sum_{pa(G_i)\in\delta(G_i)} Pr(pa(G_i)|ev)$. We will see how by assuming that collaborative ratings are independent given the evidence these computations can be considerably reduced.

Firstly, and with the idea of being general, we will introduce some notation: let $X_1, \ldots X_n$ be a set of n independent variables and let π_n represent any configuration of these variables. As these variables are independent $Pr(\pi_n) = \prod_{i=i}^{n} Pr(x_{i,j})$, where $x_{i,j}$ is the value that variable X_i takes in the configuration π_n.[16] Let δ_k be a combination with repetition of a subset of k variables and let $s \in \delta_k$ represent the fact that the value s belongs to the combination δ_k. Additionally, we say that δ_{k-1} is a s-reduction of δ_k, denoted by $\delta_k^{\downarrow s}$, if δ_{k-1} can be obtained by removing a value s from the combination δ_k. The following theorem shows how $Pr(\delta_n)$ can be computed recursively:

Theorem 3 *Let δ_n be any combination with repetition from the set of $X_1, \ldots, X_n$. If X_i is independent of X_j, $\forall i \neq j$, the probability associated with the combination δ_n can then be computed as*

$$Pr(\delta_n) = \begin{cases} Pr(x_{1,t}) & \textit{if } n = 1 \textit{ and } t \in \delta_1 \\ \sum_{s\in\delta_n} Pr(\delta_{n-1}^{\downarrow s})Pr(x_{n,s}) & \textit{if } n > 1 \end{cases} \tag{14}$$

Proof We know that $Pr(\delta_n) = \sum_{\pi_n\in\delta_n} Pr(\pi_n)$. Assuming independence between the variables, we have that $Pr(\pi_n) = Pr(\pi_{n-1})Pr(x_{n,s})$ where π_{n-1} is the configuration of the first $n-1$ variables in π_n and $x_{n,s}$ is the value of X_n in π_n. Grouping all the configurations with the same value for variable X_n we have that

$$Pr(\delta_n) = \sum_{s\in\delta_n} Pr(x_{n,s}) \sum_{\pi_{n-1}\in\delta_{n-1}} Pr(\pi_{n-1}) = \sum_{s\in\delta_n} Pr(\delta_{n-1}^{\downarrow s})Pr(x_{n,s}).$$

[16] It should be noted that in the case that all $Pr(X_a) = Pr(X_b), \forall a \neq b$, we will have a multinomial distribution of ratings, simplifying the estimation processes. Nevertheless, this situation should imply that all the collaborative nodes have the same rating probabilities, which is not a valid assumption in this domain.

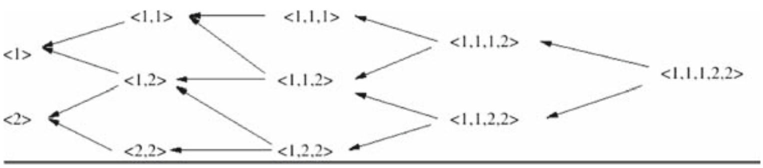


Fig. 5 Recursion graph for computing $Pr(< 1, 1, 1, 2, 2 >)$

Table 12 Algorithm for computing $Pr(\Delta)$

Computing $Pr(\Delta)$

$Pr(\delta_1) = Pr(X_1)$
for ($k = 1; k < n; k{+}{+}$)
  for each $\delta_k \in CR^r_k$ do // each combination of size k
    for ($s = 1; s <= r; s{+}{+}$) // values of X_{k+1}
      $Pr(\delta_{k\cup s}) += Pr(\delta_k) \times Pr(x_{k+1,s})$

A first idea would be to apply this result directly in order to compute $Pr(\delta(G_i)|ev)$. For instance, Fig. 5 shows the recursion graph for the computation of $Pr(< 1, 1, 1, 2, 2 >)$, where each different combination obtained after a reduction has been displayed only once. The key observation is that the number of (sub)combinations obtained after applying a reduction process is relatively small. A recursive algorithm may therefore encounter each one many times in different branches of its recursion graph. For example, Fig. 5 shows that the (sub)combination $Pr(< 1, 1, 2 >)$ should be computed twice and the (sub)combination $Pr(< 1, 1 >)$ three times. Moreover, some of these subproblems might also appear when computing different joint probabilities, such as $Pr(< 1, 1, 2, 2, 2 >)$. Applying Theorem 3 directly therefore involves more work than necessary.

We propose that every probability for a given subcombination be computed just once and its values saved in a table, thereby avoiding the work of recomputing this probability every time the subcombination is encountered.

The following algorithm (see Table 12) shows how to compute the joint probability distributions for all the possible combinations with replacement in the set Δ. We follow a bottom-up approach where we first compute the probabilities associated with the smallest (sub)combinations in terms of the number of variables used to form the combinations with repetition, and these probabilities will be used as the basis for calculating the largest combinations. Initially, when considering the first variable X_1, we have r different combinations with replacement, one for each possible value of the variable X_1. In a general stage, we then found that the probabilities associated with each combination δ_k of the first k variables are used in the computation of the probabilities of r different combinations with size $k+1$, one for each possible value of the variable X_{k+1}. Each of these combinations will be denoted by $\delta_{k\cup s}$ with $1 \leq s \leq r$.

An inspection of the algorithm yields a running time of $T(n) = \sum_{i=1}^{n} rCR^r_i$, i.e. $T(n) \in O(rn^r)$, which is much more efficient than applying the recursive algorithm from Theorem 3 directly. For example, in the case of bivaluated variables (as is usual in decision problems), we have a quadratic algorithm for combining the output of the different individuals. With respect to the memory needed to store the intermediate results, we find that the values in stage k are only used in stage $k-1$, and therefore the memory used is on the order of $O(CR_n^r)$.

References

Abellán, J., Masegosa, A.: Combining decision trees based on imprecise probabilities and uncertainty measures. In: Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Springer LNAI 4724, pp. 512–523. Hammamet, Tunisia (2007)

Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)

Albrecht, D., Zukerman, I.: Special issue on statistical and probabilistic methods for user modeling. User Model. User-Adapt. Interact. **17**(1–2) (2007)

Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: INTRIGUE: personalized recommendation of tourist attractions for desktop and handset devices. Appl. Artif. Intell. **17**(8), 687–714 (2003)

Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52. Madison, WI, USA (1998)

Butz, C.: Exploiting contextual independencies in web search and user profiling. In: Proceedings of World Congress on Computational Intelligence, pp. 1051–1056. Honolulu, HI, USA (2002)

Chen, Y.-L., Cheng, L.-C., Chuang, C.-N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. **34**, 2082–2090 (2008)

Clemen, R.T., Winkler, R.L.: Combining probability distributions from experts in risk analysis. Risk Anal. **19**, 187–203 (1999)

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: Generalizing e-Bay.NET: an approach to recommendation based on probabilistic computing. In: 1st Workshop on Web Personalization, Recommender Systems and Intelligent User Interface, pp. 24–33. Reading, UK (2005)

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: A Bayesian network approach to hybrid recommending systems. In: Eleventh International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 2158–2165. Paris, France (2006)

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F.: A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets Syst. **159**(12), 1554–1576 (2008)

Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Wiley, New York (1973)

Genest, C., Zidek, J.: Combining probability distributions: a critique and annotated bibliography. Stat. Sci. **1**(1), 114–148 (1986)

Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., Kadie, C.: Dependency networks for inference, collaborative filtering, and data visualization. J. Mach. Learn. Res. **1**, 49–75 (2001)

Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: SIGIR 99: Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237. University of California, Berkeley, USA (1999)

Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. **22**(1), 5–53 (2004)

Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: AVI ’04: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 48–54. New York, NY, USA (2004)

Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization, pp. 596–627. Springer Verlag LNCS 4321, (2007)

Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 226–238 (1998)

Kudenko, D., Bauer, M., Dengler, D.: Group decision making through mediated discussions. In: Proceedings of the Ninth International Conference on User Modeling (UM 03), Springer LNAI 2702. University of Pittsburgh, Johnstown, USA (2003)

Lekakos, G., Giaglis, G.M.: A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. User Model. User-Adapt. Interact. **17**(1–2), 5–40 (2007)

Masthoff, J.: Group modeling: selecting a sequence of television items to suit a group of viewers. User Model. User-Adapt. Interact. **14**, 37–85 (2004)
Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Model. User-Adapt. Interact. **16**, 281–319 (2006)
McCarthy, J.E., Anagnost, T.D.: 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, p. 348. New York, NY, USA (2000)
Miyahara, K., Pazzani, M.J.: Collaborative filtering with the simple Bayesian classifier. In: Pacific Rim International Conference on Artificial Intelligence, pp. 679–689. Melbourne, Australia (2000)
Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: DL '00: Proceedings of the Fifth ACM Conference on Digital libraries, pp. 195–204. New York, NY, USA (2000)
O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: PolyLens: a recommender system for groups of user. In: ECSCW'01: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Bonn, Germany (2001)
Pennock, D.M., Wellman, M.P.: Graphical models for groups: belief aggregation and risk sharing. Decis. Anal. **2**(3), 148–164 (2005)
Popescu, A., Ungar, L., Pennock, D., Lawrence, S.: Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: 17th Conference on Uncertainty in Artificial Intelligence, pp. 437–444. Seattle, Washington, USA (2001)
Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40**(3), 56–58 (1997)
Schiaffino, S.N., Amandi, A.: User profiling with case-based reasoning and Bayesian networks. In: IBERAMIA-SBIA 2000 Open Discussion Track, pp. 12–21. Atibaia, So Paulo, Brazil (2000)
Yu, Z., Zhou, X., Hao, Y., Gu, J.: TV Program recommendation for multiple viewers based on user profile merging. User Model. User-Adapt. Interact. **16**(1), 63–82 (2006)
Zukerman, I., Albrecht, D.: Predictive statistical models for user modeling. User Model. User-Adapt. Interact. **11**, 5–18 (2001)

Authors' vitae

Luis M. de Campos born in 1961, received his M.S. degree in Mathematics in 1984. He completed his PhD Thesis in 1988 and became Lecturer in Computer Science in 1991 at the University of Granada, Spain. He is currently Full Professor at the Department of Computer Science and Artificial Intelligence at the University of Granada. His current areas of research interest include Probabilistic Graphical Models, Machine Learning and Information Retrieval.

Juan M. Fernández-Luna received his Computer Science degree in 1994 at the University of Granada, Spain. In 2001 he got his PhD at the same institution, working on a thesis in which several retrieval models based on Bayesian networks for Information Retrieval where designed. This subject is still his current research area, although moving to structured documents. He has got experience organising international conferences and has been guest editor of several special issues about Information Retrieval. He is currently assistant professor at the Department of Computer Science and Artificial Intelligence at the University of Granada.

Juan F. Huete born in 1967, received his M.S. degree in Computer Science in 1990. He completed his PhD Thesis on "Learning Belief Networks: Non-Probabilistic Models" in 1995 and became Lecturer in Computer Science in 1997 at the University of Granada, Spain. His current areas of research interest are Probabilistic Graphical Models and its application to Information Retrieval, Recommender Systems and Text Classification.

Miguel Angel Rueda-Morales received his B.A. in Computer Science in 2003 from the University of Granada, Spain, and is currently a PhD candidate at its Department of Computer Science and Artificial Intelligence. His primary interests lie in the area of recommender systems using probabilistic models. His contribution is based on experiences gained both from his Ph.D. work as well as his current research on recommender systems.