Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 159 (2008) 1554–1576
FUZZY sets and systems
www.elsevier.com/locate/fss

# A collaborative recommender system based on probabilistic inference from fuzzy observations☆

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete*

*Departamento de Ciencias de la Computación e Inteligencia Artificial E.T.S.I. Informática y Telecomunicaciones, Universidad de Granada, 18071 Granada, Spain*

Available online 26 January 2008

**Abstract**

The problem of building recommender systems has attracted considerable attention in recent years. The objective of this paper is to automatically suggest and rank a list of new items to a user based on the past voting patterns of other users with similar tastes. The proposed model can be considered as a Soft Computing-based collaborative recommender system.

The combination of Bayesian networks, which enables an intuitive representation of the mechanisms that govern the relationships between the users, and the Fuzzy Set Theory, enabling us to represent ambiguity or vagueness in the description of the ratings, improves the accuracy of the system.


*Keywords:* Collaborative recommender system; Bayesian networks; Fuzzy observations

## 1. Introduction

Over the last 10 years, web use has changed from academic use to commercial use and this has meant that an enormous amount of available information is not accessible for the users, because, for example, they are unaware that it exists. This situation offers a very attractive framework for researching in the form of new accurate and efficient techniques designed to access this information. In this framework, *recommender systems* (*RS*) have emerged to help people deal with this information overload. Broadly speaking, an RS provides specific suggestions about items (or actions) within a given domain, which may be considered of interest to the user [36]. Examples of such applications include recommending books, CDs and other products at Amazon.com [27], movies by MovieLens [28], books at LIBRA [31], electronic television program guides [32], etc.

One of the inputs for an RS is a database of user preferences which is known as the *user profile*. This profile has been provided either explicitly (by means of a form or a questionnaire on logging in) or implicitly (using purchase records, viewing or rating items, visiting links, taking into account membership to a certain group, etc.). These preferences will be collected over time, giving a much more reliable identification of user preferences or tastes. For instance, in MovieLens, users can rate any of the movies they have seen using a range in the set $\{1 = Awful, 2 = Fairly\ Bad, 3 = OK, 4 = Enjoyable, 5 = Must\ see\}$ by means of forms such as the one presented in Fig. 1.

☆ This work has been supported by the Spanish 'Ministerio de Educación y Ciencia' and 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía'under Projects TIN2005-02516 and TIC-276, respectively.
* Corresponding author.
*E-mail address:* jhg@decsai.ugr.es (J.F. Huete).

0165-0114/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2008.01.016

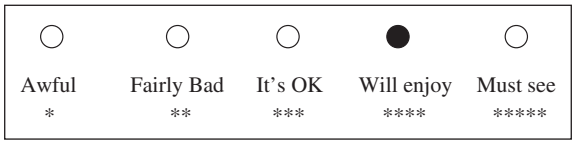


Fig. 1. Rating process.

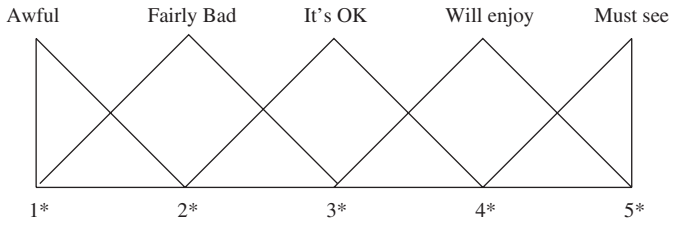


Fig. 2. Rating process.

There are several types of RS, classified according to the information used when recommending. This paper focuses on the variant called *collaborative filtering*, which attempts to identify groups of people with similar tastes to the user's and to recommend items that they have liked. In this case, the objective is usually to predict the utility of an unseen item for an active user based on items previously rated by other users. Continuing with the MovieLens example, given an unseen movie, the output of the system might also be a label from *Awful* to *Must see*, representing the predicted degree of user satisfaction.

More specifically, we propose to use two main Soft Computing techniques in order to model the uncertainties and the tolerance of imprecision related to the recommending process. On the one hand, we propose to use the *Bayesian network* (*BN*) [34] formalism to model the way the user's ratings are related. We believe that it is essentially a statistical process and by using BN we could combine a qualitative representation of how users and items are related (explicitly representing dependence and independence relationships in a graphical structure) as well as a quantitative representation by means of a set of probability distributions, measuring the strength of these relationships.

On the other hand, it must be pointed out that fuzziness exists in the rating process. Rating an item usually implies the selection of a vote from a set of labels. For example, in Fig. 1, the user is expressing the fact that *My rating for this movie is* 'Will enjoy'. However, there is often no meaningful way to set the boundary between two consecutive labels such as 'Will enjoy' and 'Must see'. For this reason, we shall also explore the advantages of considering the rating alternatives as vague concepts. Fuzzy sets formalize the idea of graded membership of an element to a given set. Following Zadeh's [47] definition, a fuzzy set $A$ of a reference set $\Omega$ is identified by its membership function, $\mu_A : \Omega \longrightarrow [0, 1]$, where $\mu_A(x)$ is interpreted as the membership degree of element $x$ in the fuzzy set $A$, $\forall x \in \Omega$. Fig. 2 represents a possible fuzzy definition of the set of rating alternatives in MovieLens.

Our approach for modeling RSs therefore involves processing two different types of uncertainty: probability arising from a lack of knowledge of how the different users are related, and fuzziness concerned with the ambiguity or vagueness in the description of the ratings. To the best of our knowledge, no study has tried to combine these two Soft Computing techniques for this purpose. We shall study how the combination of both theories leads to an improvement when modeling a collaborative-based RS.

The second section of this paper presents the fundamentals of BN and third section present the fundamentals of the recommending problem. Section 4 details some of the work published on RS, focusing on both probabilistic and fuzzy approaches. Section 5 describes the information sources and Section 6 presents the BN topology, its learning algorithm and the estimation of the conditional probability distributions from the data sets of fuzzy observations. Section 7 discusses how the recommendations can be computed, i.e. how inference is performed and how the most appropriate fuzzy label is selected. Section 8 presents some experimental results, and finally Section 9 includes the conclusions and some remarks about further research.

## 2. A summary of BNs

Probability Theory provides a rigorous formalism for modeling and analyzing random phenomena and concerns how probabilities may be assigned to (precise) events and how these probabilities can then be manipulated in order to obtain new conclusions (probabilistic reasoning). In recent decades, BNs [34] have become one of the most consolidated methodologies for probabilistic inference. They are graphical models capable of efficiently representing and manipulating $n$-dimensional probability distributions by combining a qualitative and a quantitative representation of the problem by means of:

- a *directed acyclic graph* (DAG), $G = (V, E)$, where the nodes in $V$ represent the random variables from the problem we want to solve, and the topology of the graph (the arcs in $E$) encodes dependence relationships between the variables. In addition, the absence of an arc between any two nodes represents an independence relationship between the variables.
- a set of conditional probability distributions drawn from the graph structure: for each variable $X_i \in V$ we have a family of conditional probability distributions $\Pr(X_i|pa(X_i))$, where $pa(X_i)$ represents any combination of the values in the parent set of $X_i$ in $G$, $Pa(X_i)$.

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

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

where $pa(x_i)$ are the values taken by the parent set of $X_i$ in $c$. This decomposition of the joint distribution results in important savings in storage requirements.

In a probabilistic framework, the user usually has some evidence of the state that a variable (or a set of variables) takes, and the problem is to compute the conditional (a posteriori) probability distribution for a variable given the evidence, $\Pr(X_i|ev)$. Given the joint distribution in Eq. (1), all the possible inference queries can be answered by marginalization (summing out over irrelevant variables). One of the main advantages of a BN is that by using the independencies represented in a BN, this computation can be performed efficiently by instantiating the evidence and performing a propagation (probabilistic inference) process through the network [34].

## 3. Recommender systems

RSs emerged as an independent research area in the mid-1990s when researchers started focusing on recommendations that explicitly relied on the rating structure. For example, in a movie recommending application (such as the one at MovieLens [28]), users initially rate a subset of movies that they have already seen. The usual formulation of the problem is then to predict the vote or rate that an active user should give to an unseen item. This estimation could be used to recommend to the user those items with the highest estimated ratings. Many different approaches to the RS have been published [22,1], using methods from machine learning, approximation theory and various heuristics. Independently of the technique being used, RSs are usually classified into the following categories, based on how the recommendations are made:

- *Content-based RSs* store content information about each item to be recommended. This information will be used to recommend items similar to the ones the user preferred in the past, based on how similar certain items are to each other or the similarity with respect to user preferences (also represented by means of a subset of content features). The main problems with content-based RS are firstly, the difficulty of making accurate recommendations to users with very few ratings, and secondly, overspecialization since the system would recommend similar items to those already rated.
- *Collaborative filtering RSs* attempt to identify groups of people with similar tastes to those of the user and recommend items that they have liked. According to [3], collaborative RS can be grouped into *memory-based* and *model-based* approaches. On the one hand, memory-based algorithms use the whole rating matrix to make recommendations. In order to do so, they use some kind of aggregation measure considering the ratings of other (most similar) users for the

same item. Different models can be obtained by considering different similarity measures and different aggregation criteria. The most common approach is

$$r_{a,j} = k \sum_{U_i \in \mathcal{U}_a^*} \text{sim}(U_a, U_i) \times r_{i,j}, \tag{2}$$

where $r_{i,j}$ denotes the rating given by user $U_i$ to the item $I_j$, $k$ is a normalizing factor, sim is a similarity (distance) measure between users and $\mathcal{U}_a^*$ denotes the set of users that are most similar to $U_a$. On the other hand, in model-based algorithms the predictions are made by building (offline) an explicit model of the relationships between items. This model is then used (online) to finally recommend the product to the users. In this approach, the predictions are therefore not based on any ad hoc heuristic, but rather on a model learnt from the underlying data using statistical and machine learning techniques.

Collaborative RSs have their own limitations: firstly, there is the new user problem since it is necessary for a set of items to be rated in order to perform similarity analysis, and the greater the number of ratings performed by a user, the more accurate the assignment to a group of similar users; and secondly, there is the new item problem since an item which has not been rated previously cannot be recommended.

- *Hybrid RS* combine both content and collaborative approaches. Different combination methods could be used from running both alternatives separately and combining their prediction to construct a general unifying model which incorporates both models.

## 4. Related work

Since our approach to model the collaborative-based RS problem is an hybrid of two well known Soft Computing techniques, on the one hand Probability Theory (by means of BN) and, on the other hand Fuzzy Sets Theory, in this section we will only discuss in more detail those models based on probabilistic or fuzzy logic formalisms. A more general classification of RS research considering the techniques and the approach used to perform recommendations can be found in [1].

### 4.1. Probabilistic-based RSs

By focusing on a probabilistic approach to RS, many variants can be found. Generally, and in order to model the recommending problem with BN it is necessary to learn the structure from a set of observed data. There are two main approaches for finding the structure. The first approach poses learning as a constraint satisfaction problem. By estimating properties of dependences or independences among the attributes in the data, a network that exhibits the underlying relationships is built. The second approach poses learning as an optimization problem. By means of a statistically motivated score, a search for an optimal structure that is optimum to the observed data is performed.

Thus, focusing on *content-based RS*, learning as constraint satisfaction problem is posed at [4], where the user profile is learnt considering contextual independences. Also, by assuming independence between variables, Bayesian classifiers have been used in [31,33] to estimate the probability that an item belongs to a certain class (relevant or irrelevant) given the item description. Finally [5,6] model the item descriptions with a BN and estimate the probability that a user rates an item with a value given the user preferences (also represented by means of a subset of content features). In [21] learning is posed as an optimization problem and applies a Branch and Bound methodology to search for the structure optimizing the observed data with an scoring criterion and [41] uses learning algorithms to built a user profile by combining BNs with case-based reasoning techniques.

Considering *collaborative RS*, we can distinguish between the two learning approaches. In the first case, considering collaborative recommendation as classification problem, the full joint probability distribution capturing the data is obtained by means of a mixture of conditional independence models. This kind of model range from the classical Naive-Bayes algorithm to more sophisticated techniques [3,20,29,37]. The second approach builds several models (one for each user represented as a probabilistic decision tree learnt using a statistical score) and these predict the likelihood of an individual item given a combination of the observed votes for the other users [18].

Finally, [35] is an example of a *probabilistic hybrid RS*, which has been proposed to facilitate online document browsing. In this model, a joint density function is constructed assuming the existence of a hidden variable representing the different topics of the documents. This variable renders users, documents and words conditionally independent. Additionally in [7], a BN-based model is built which incorporates both content-based and collaborative characteristics. Also a Bayesian hierarchical modeling approach has been used to find a user profile by combining user-specific (content) and shared (collaborative) information components [49].

### 4.2. Application of fuzzy sets to RSs

Fuzzy sets also provide an appropriate theoretical foundation for several kinds of RS extensions and developments. Thus, Yager [45] gives a methodology for using fuzzy set methods to represent the content knowledge in a content-based RS. Dubois et al. [15] also discuss how fuzzy sets can be used as the basis for case-based decision support processes in order to represent user preferences in a flexible way. A different approach is to use fuzzy values to represent the attributes of a item. In this sense, [42] use a fuzzy classification to determine the value of the product. This values could be used to influence (re-ranking) the output of a standard RS.

More practical approaches can also be found in the literature: in terms of content-based RS, fuzzy logic is applied to guide users when buying consumer electronics [8]; the system has specific domain knowledge and is able to interact with the consumer. Fuzzy clustering methods [2,46] which classify items into more than one cluster have been used to provide a more realistic representation of the content profiles [10]. In the case of collaborative RS, fuzzy clustering has also been used to develop collaborative RS by grouping together users with similar tastes [23] or performing item clustering [26]. Finally, in [9] an agent-based collaborative RS is developed. In this case, each agent express their recommendations as fuzzy sets, and the corresponding collaborative filtering algorithm becomes an aggregation of fuzzy sets (by means of a weighted arithmetic mean) taking into account the circumstance in the past.

## 5. Information source

Before presenting our model, we shall briefly discuss the information source. Typically, we have a large number $m$ of items $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$, a large set of $n$ users, $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ and for each user, a set of ratings about the quality of certain observed items in $\mathcal{I}$. In this paper, we will assume that the user's ratings (preferences) are collected by means of expressions such as '*this item is rated with S*', with $S$ taking its values on some finite set $\mathcal{S}$.

In practice, however, it might be difficult for the user to consider crisp values over $\mathcal{S}$ when rating an item. For example, a user might prefer to indicate that 'This item is excellent' rather than 'I would rate this item 18 (out of 20)'. It therefore seems natural to use fuzzy set formalism to describe the user's degree of satisfaction obtained after observing an item. In this case, users could state their (imprecise) information in the form of fuzzy labels. For instance, if we interpreted the set of ratings used in MovieLens ({*Awful, Fairly Bad, OK, Enjoyable, Must see*}) as fuzzy labels, rating a movie as *Must see* does not necessarily rule out the possibility of it being graded as *Will enjoy*.

Although users might think in terms of vague concepts when rating, it is quite common that, in the end, an RS stores their ratings using crisp values, $s_i \in \mathcal{S}$, with the consequent loss of information (for instance, MovieLens uses the set $\mathcal{S} = \{1, 2, \ldots, 5\}$). In our opinion, a kind of mapping from the fuzzy concepts to the set of discrete labels has been used, without considering the vagueness in the user's rating process. Continuing the MovieLens example, whenever a user rates a film as *Must see*, the system stores its associated integer rate 5. However, if we assume that *Must see* might be described with the fuzzy set $\{0/1, 0/2, 0/3, 0.5/4, 1/5\}$, this excludes the possibility that the user's satisfaction degree could be more or less described by means of the value 4 (representing *Will enjoy*).

In this paper, we shall study the advantages of considering the user rates as fuzzy observations of $\mathcal{S}$, i.e. each particular rate is considered as a fuzzy subset of $\mathcal{S}$. We use $\mathcal{S}_L = \{l_1, l_2, \ldots, l_r\}$ to denote the set of fuzzy labels used to describe an item and with an integer in the set $\mathcal{S} = \{1, 2, \ldots, r\}$ their respective associated crisp values, with $r$ being the number of different fuzzy labels used to rate an item.

The set of observed data can be viewed as a very sparse $n \times m$ matrix, $\mathbf{R}$, since a typical user only rates a very small fraction of the items. In the matrix, $\mathbf{R}[a][j]$ represents the rate of user $U_a$ for the item $I_j$ and will also be denoted as $l_{a,j}$. This is assumed to be zero when the item has not been rated (observed) by the user. An example of a user-item rating matrix is presented in Table 1.

Table 1
Database of user rates: each rate is a label from the set $\{l_1 = \text{Awful},\ l_2 = \text{Fairly Bad},\ l_3 = \text{OK}, l_4 = \text{Will enjoy},\ l_5 = \text{Must see}\}$

| | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $I_8$ | $I_9$ | $I_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $U_1$ | $l_5$ | $l_5$ | $l_3$ | $l_1$ | $l_3$ | $l_3$ | 0 | 0 | 0 | 0 |
| $U_2$ | $l_5$ | $l_5$ | $l_4$ | $l_1$ | $l_1$ | $l_4$ | 0 | 0 | 0 | 0 |
| $U_3$ | $l_4$ | $l_4$ | $l_3$ | $l_2$ | $l_2$ | $l_4$ | 0 | 0 | 0 | $l_3$ |
| $U_4$ | $l_1$ | 0 | $l_1$ | 0 | 0 | $l_2$ | $l_1$ | $l_3$ | 0 | $l_5$ |
| $U_5$ | 0 | 0 | $l_2$ | 0 | $l_1$ | 0 | 0 | 0 | $l_3$ | 0 |
| $U_6$ | 0 | 0 | $l_5$ | 0 | $l_4$ | 0 | 0 | 0 | 0 | $l_3$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

Nevertheless, since our model is based on a probabilistic framework we need a mechanism to estimate for each (crisp) rate $s_i$, $s_i \in \mathcal{S}$, probability values from the set of imprecise (fuzzy) observations, $\mathbf{R}$. Different approaches to tackle this problem have been published [43,44]. In this paper, we propose the use of a function $\phi$ which measures the adaptation of each element $s_i$ to the fuzzy observation $l$, with $l \in \mathcal{S}_L$. Following [17], this adaptation $\phi$ can be interpreted as the uncertainty degree of $s_i$ given the observation and should have the properties of an uncertainty measure. We can consider two different ways to compute this function that might be interpreted as a probability distribution on $\mathcal{S}$:

(i)

$$\phi_1(s_i|l) = \frac{\mu_l(s_i)}{\sum_k \mu_l(s_k)}. \tag{3}$$

(ii)

$$\phi_2(s_i|l) = \begin{cases} \dfrac{1}{t} & \text{if } \mu_l(s_i) = 1, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

with $t$ being the number of times that the $\mu_l(s_i) = 1$.

For example, given the fuzzy observation $l =$ 'Must see' defined with the certainty degree $\{0/1, 0/2, 0/3, 0.5/4, 1/5\}$ we obtain the following $\phi$ functions: $\phi_1(s_i|l) = (0, 0, 0, 0.333, 0.666)$ and $\phi_2(s_i|l) = (0, 0, 0, 0, 1)$.

Once we have defined the $\phi$ functions, it is easy to estimate a probability distribution associated with a set of $n$ fuzzy observations $l_1, \ldots, l_n$ by means of the following expression:

$$\Pr(\text{vote} = s_i|l_1, \ldots, l_n) = \frac{\sum_{j=1}^{n} \phi(s_i|l_j)}{n}. \tag{5}$$

It should be noted that if the set of fuzzy labels $\mathcal{S}_L$ represents a triangular full fuzzy covering of the domain (for each $s_i$ there is a triangular label $l$ which $\mu_l(s_i) = 1$), then the probabilities obtained using $\phi_2$ in Eq. (5) coincide with those obtained using a frequentist approach for estimating the probabilities.

## 6. BN-based collaborative RS

When we are interested in representing our knowledge by means of BNs, the first task is to select those variables which are relevant to the problem we are tackling. Each variable will be represented as a node in the DAG and whenever two variables are related, a path must exist between them in the graph. These connections can be determined from an input data set by means of a learning algorithm.

In our case, we must model both the relation $\mathcal{I} \longrightarrow \mathcal{U}$ by modeling the database of user votes for the set of observed items and also the relation $\mathcal{U} \longrightarrow \mathcal{U}$ by modeling the relationships between users. Therefore, at this preliminary stage, we will consider the set of items $\mathcal{I}$ and the set of users $\mathcal{U}$ as variables in the BN (nodes in the graph).

Our first objective will be to model the user's voting pattern which represents the dependence relationships between items, $\mathcal{I}$, and user votes, $\mathcal{U}$. In this case, it is clear that the voting pattern of each user ($U_a$) will depend directly on the

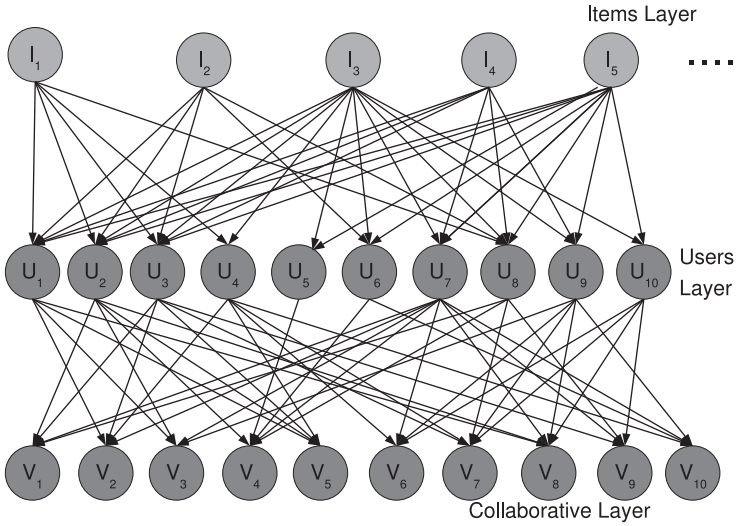


Fig. 3. Collaborative Recommending System topology.

Table 2
Most probable fuzzy label

| $\mathcal{S}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\Pr(V_a = s\|ev)$ | 0.20 | 0.25 | 0.2 | 0.3 | 0.05 |
| $\mathcal{S}_L$ | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ |
| $\Pr_L(V_a$ rate is $l_s\|ev)$ | 0.325 | 0.45 | 0.475 | 0.425 | 0.2 |

vote given to each observed item. We will therefore include an arc from each item, $I_j$, voted by user $U_a$ to the node representing that user (the upper part of Fig. 3 shows the relations expressed in Table 2).

In relation to the states (the domain) of the variables:

- Each item $I_j \in \mathcal{I}$ will have an associated random binary variable, taking values from the sets $\{i_{j,0}, i_{j,1}\}$, respectively. The subindex 0 is used to represent the fact that the item is not-relevant to the user's interest whereas the subindex 1 represents the relevance alternative.
- Each user variable $U_a \in \mathcal{U}$ will store the probability distribution associated to their rating pattern, i.e. information about the likelihood that $U_a$ could vote with value $i$, (vote $U_a = i$), with $i \in \{0, 1, 2, \ldots, r\}$. Each user variable will therefore take values in $\{u_{a,1} \ldots, u_{a,r}\} \cup \{u_{a,0}\}$. It should be noted that a new state ($u_{a,0}$) has been added in order to represent the situation where the user has no interest in voting.

In a collaborative RS, the prediction of the vote for a given user depends on the votes of people with similar tastes or preferences. Our model therefore might be able to represent relations between users, $\mathcal{U} \longrightarrow \mathcal{U}$. Regardless of the mechanism used to find these relationships, they should be modeled in the BN by the inclusion of arcs between any two related users. Thus, whenever a dependence (similarity) between the preferences of user $U_a$ and user $U_b$ has been found, an arc connecting both nodes should be included in the BN. However, taking into account that similarities between users' tastes tend to be symmetric (when $U_a$ is highly related with $U_b$, it is also common for $U_b$ to be related with $U_a$), a cycle could be included in the BN, which is forbidden in a BN topology.

In order to facilitate the presence of these relationships in the model, we therefore propose that a new set of nodes $\mathcal{V}$ be included to denote collaborative votes. There is one collaborative node for each user in the system, i.e. $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$. Following the performance of a collaborative RS, these nodes will also be used to predict the vote that the active user could give to an unseen item and they will therefore take their values in the set of valid rating labels, i.e. $\{v_{a,1}, v_{a,2}, \ldots, v_{a,r}\}$, omitting $v_{a,0}$ as an alternative state.

### 6.1. Learning stage

BN learning involves searching through different network structures. This is exponential to the number of variables. In order to make this stage efficient, we follow a constraint satisfaction learning approach. Particularly, by considering some (in)dependence restrictions on the way the variables are related (the links in the graph), we reduce considerably the time needed to learn the model. We would like to say that with the resulting model the final recommendations are obtained as a mixture of a set of conditional dependent models, in opposition to the models obtained by posing the recommendation as a classification task [3,20,29,37], where the recommendations are obtained as a mixture of conditional independent models.[1]

The parent set of a collaborative variable $V_a$ in the graph, $Pa(V_a)$, will be learnt from the database of votes, $\mathbf{R}$. This set will contain those user variables, $U_b \in \mathcal{U}$, with $U_a$ and $U_b$ having the greatest similarity between their tastes, and constituting the collaborative layer of the BN (see the lower part of Fig. 3). Thus, given a similarity measure, the set $Pa(V_a)$ should be obtained by using a threshold or considering only the first $p$ variables with the highest similarity.

Given a pair of users, a first idea for measuring the similarity between their voting pattern would be to use Pearson's correlation coefficient (PCC), a criterion which is normally used as the basis for calculating the weights in different collaborative systems.[2] In this case, the parent set of the active collaborative node $V_a$ could be obtained by selecting those users, $U_b$, with the highest absolute PCC value:

$$PCC(U_a, U_b) = \frac{\sum_j (r_{a,j} - \bar{r}_a)(r_{b,j} - \bar{r}_b)}{\sqrt{\sum_j (r_{a,j} - \bar{r}_a)^2 \sum_j (r_{b,j} - \bar{r}_b)^2}}, \tag{6}$$

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

From our point of view, various problems arise when this correlation coefficient is applied to our domain, where the data set R is a very sparse matrix. For example, considering the data in Table 1, $PCC(U_4, U_5) = 1$ since both users only rated one common item. In this case, $U_4$ will be set as the parent of $V_5$ and also $U_5$ is a parent of $V_4$, resulting in low quality parent sets.

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

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

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

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

$$sim(U_a, U_b) = D(U_a, U_b) \times abs(PCC(U_a, U_b)), \tag{7}$$

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

---

[1] Note that in many real situations this assumption that, given the class variable, the set of attributes are independent does not necessary holds.

[2] The use of the cosine measure [38] has also been explored, but with no improvement in system performance.

### 6.2. Estimating conditional probability distributions

In order to complete the model's specification, the numerical values for the conditional probabilities must be estimated from the data sets, but prior to this, we will introduce some notation. Consequently, given a variable $X_i$, lowercase letters are used to denote the carrying out of the variables. For instance, $x_{i,j}$ denotes the fact that variable $X_i$ takes the $j$th-value. We write $\Pr(x_{i,j}|pa(X_i))$ for $\Pr(X_i = x_{i,j}|pa(X_i))$, with $pa(X_i)$ denoting a configuration of the parent set of $X_i$, $Pa(X_i)$, or sometimes $\Pr(X)$ to denote the probability distribution.

We must distinguish between the set of items $\mathcal{I}$ and the sets of user and collaborative nodes, $\mathcal{U}$, $\mathcal{V}$. In the first case, since they are root nodes in the graph, they store marginal probability distributions which are linear in size to the number of states, whereas in the second case, these variables must store a set of conditional probability distributions with an exponential size to the number of parents. Since a user can rate a large number of items and a collaborative node might be related to a great number of users, the assessment and storage of these probability values can be quite complex. We therefore propose that a canonical model (similar to the one presented in [6]) be used to represent the conditional probabilities, thereby enabling us to design a very efficient inference procedure. Thus, for a given node $X_i$, we define these probabilities as follows:

$$\Pr(x_{i,j}|pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}), \tag{8}$$

where $y_{k,l}$ is the value that variable $Y_k$ takes in the configuration $pa(X_i)$ and $w(y_{k,l}, x_{i,j})$ is a weight measuring how this $l$th value of variable $Y_k$ describes the $j$th state of node $X_i$. The only condition we must impose is that by means of these weights a proper probability distribution can be defined (see Theorem 1, Section 7.1).

We should present some guidelines about how these probability values might be estimated:

- For every item node, $I_j$, we need to assess its a priori probability of relevance. Since all the items are equally probable a priori, in this paper we propose that a constant value $\alpha$ be used, i.e. $\Pr(i_{j,1}) = \alpha$ and $\Pr(i_{j,0}) = 1 - \alpha$, $\forall I_j \in \mathcal{I}$.
- For every user node $U_k$, we need to assess a set of conditional probability distributions, one for each possible configuration of its parent set, i.e. the set of items rated by $U_k$. These probabilities will represent the rating pattern for user $U_k$ and considering the above restrictions, they will be computed using a canonical model (see Eq. (8)). In this case, assuming that the user $U_k$ rated an item $I_j$ with the label $l$ (it should be noted that in this case each vote is considered as a fuzzy observation, i.e. $l \in \mathcal{S}_L$), these weights could be defined by means of

$$w(i_{j,0}, u_{k,s}) = 0 \ \ \forall s \neq 0, \quad w(i_{j,0}, u_{k,0}) = 1/|Pa(U_k)|,$$

$$w(i_{j,1}, u_{k,s}) = \frac{\phi(s|l)}{|Pa(U_k)|} \ \ \forall s \neq 0, \quad w(i_{j,1}, u_{k,0}) = 0. \tag{9}$$

  For instance, considering the data in Table 1, the adaptation function $\phi_1$ and the set of triangular fuzzy labels defined by means of

$$\mu_{l_i}(s_j) = \begin{cases} 1 & \text{if } i = j, \\ 0.5 & \text{if } abs(i-j) = 1, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

  then $w(i_{1,1}, u_{1,\bullet}) = (0, 0, 0, 0, 0.083, 0.166)$, with the $k$th value in this tuple being the weight $w(i_{1,1}, u_{1,k})$, $k = 0, \ldots, 5$ and $w(i_{3,0}, u_{1,\bullet}) = (0.166, 0, 0, 0, 0, 0)$. Then, given the configuration $c = \{i_{1,1}, i_{2,1}, i_{3,0}, i_{4,1}, i_{5,0}, i_{6,0}\}$, $\Pr(U_1|c) = (0.5, 0.111, 0.055, 0.0, 0.111, 0.222)$. In addition, using $\phi_2$, $\Pr(U_1|c) = (0.5, 0.166, 0.0, 0.0, 0.0, 0.333)$.
- Focusing on collaborative nodes $\mathcal{V}$, for each node $V_a$ we must compute those weights $w(u_{b,\bullet}, v_{a,\bullet})$ given by users $U_b$ with similar tastes, i.e. $U_b \in Pa(V_a)$. We propose that a double weighting scheme be used: on one side, and in view of the fact that user ratings are related statistically, it can be considered that these weights should depend on the frequency that user $U_a$ votes with value $s$ given that user $U_b$ has the state $t$, i.e. $freq(u_{a,s}|u_{b,t})$, and on the other, considering that stronger weights should be assessed to the most similar users, it seems natural that these weights

will also depend on the similarity degree between users. The way the weight associated to user $U_b$ is distributed is therefore defined by means of the following equation:

$$w(u_{b,t}, v_{a,s}) = \frac{freq(u_{a,s}|u_{b,t}) \times sim(U_a, U_b)}{NF(V_a)} \tag{11}$$

with $s \in \mathcal{S}$, $t \in \mathcal{S} \cup \{0\}$ and $NF(V_a)$ being a normalization factor defined as

$$NF(V_a) = |Pa(V_a)| \sum_{U_b \in Pa(V_a)} sim(U_a, U_b).$$

With respect to the estimation of $freq(u_{a,s}|u_{b,t})$, we must discuss two different situations. The first of these is when user $U_b$ has no interest in voting ($U_b = u_{b,0}$). In this case, we propose that an unbiased weighting scheme be used, i.e. $\Pr(u_{a,s}|u_{b,0}) = 1/r$, for all $s \in \mathcal{S}$. With this criterion, we represent the fact that the weight associated with the 'no interest in voting' situation will be distributed uniformly among the different candidate rates at collaborative nodes. The second situation is related to the real voting alternatives for user $U_b$, i.e. $freq(u_{a,s}|u_{b,t})$ with $t \neq 0$, that can be estimated from the data sets by means of the following expression:

$$freq(u_{a,s}|u_{b,t}) = \frac{N^*(u_{b,t}, v_{a,s}) + \beta q_s}{N^*(u_{b,t}) + \beta} \quad 1 \leqslant t, \quad s \leqslant r,$$

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

## 7. Computing the recommendations

In order to study how to perform recommendations, it is previously necessary to discuss how the users should interact with the system. For instance, following the example of MovieLens, let us assume that user $U_a$ has not seen the film '*Finding Nemo*'. The RS objective is to predict the user's satisfaction degree after seeing this movie. Naturally, the system output might be a rating expressed as a single fuzzy label in a range from *Awful* to *Must see*. This prediction, following the typical performance of a collaborative RS, should depend on the particular vote that users with similar tastes gave to this movie.

In order to achieve this objective, we propose a two-step process: firstly, and in view of the interest in the unobserved item as the evidence, we shall compute the a posteriori probability distribution for the collaborative node $V_a$, $\Pr(V_a = s|ev)$ for all $s \in \mathcal{S}$, (see Section 7.1) and secondly, given this information we must select the fuzzy label $l$, with $l \in \mathcal{S}_L$, better predicting the user's satisfaction degree. This second step involves a mapping from a probability over $\mathcal{S}$ to the set of vague concepts $\mathcal{S}_L$ (see Section 7.2).

It should be noted that the system could also be used to perform other related tasks: *finding good items* attempting to suggest specific items to the active user, giving it a ranked list of recommended items or the 'best bet' item as the output. This can be achieved by instantiating in turn all the unseen items for the active user and ranking these items using the satisfaction degree. A different task is to *predict votes for a set of products*. This can be useful in e-commerce situations when searching for users interested in buying (observe) a pack of products.

### 7.1. Probabilistic inference

The aim of this section is to study how to compute the a posteriori probability distribution $\Pr(V_a = s|ev)$, $1 \leqslant s \leqslant r$ for a collaborative node $V_a$. The first problem that we must consider, however, is to determine which variables (nodes in the graph) are directly affected by the evidence, i.e. those variables affected when focusing on the particular unobserved item. In this case we have two possible situations:

- For any user who previously rated this item we know exactly the vote assessed by the user: 'User $U_i$ rated the item $I_j$ with label $l$'. Thus, and considering the user votes as fuzzy observations, the new probability distribution representing

his/her pattern of vote given the evidence, $\Pr(U_i = s|ev)$, coincides with the adaptation degree for each candidate rate $u_{i,s}$ to the fuzzy observation. These situations can be captured by means of the following expression:

$$\Pr(U_i = s|ev) = \phi(s|l) \quad \text{if } U_i \text{ rated } I_j \text{with label } l. \tag{12}$$

In order to illustrate this situation, continuing with the above example, let us suppose that we are interested in '*Finding Nemo*' and let $U_i$ be a user who rated this movie with *Must see*. In this case, considering $\phi_1$ and the fuzzy label *Must see* $= \{0/1, 0/2, 0/3, 0/4, 1/5\}$, we have that $\Pr(U_i = 5|ev) = 1$ and $\Pr(U_i = s|ev) = 0, 1 \leqslant s \leqslant 4$ and considering *Must see* as the fuzzy set $\{0/1, 0/2, 0/3, 0.5/4, 1/5\}$, we have that $\Pr(U_i = 5|ev) = 0.6666$, $\Pr(U_i = 4|ev) = 0.3333$ and $\Pr(U_i = s|ev) = 0, 0 \leqslant s \leqslant 3$.

- When focusing on users who did not rate this item, we do not have any evidence of their voting pattern and it is therefore natural to consider that the a posteriori probability distribution representing their voting pattern coincides with the a priori one, i.e.

$$\Pr(U_i = s|ev) = \Pr(U_i = s). \tag{13}$$

It is interesting to note that the a priori probabilities could be pre-computed by propagating in the BN without evidence once (see Theorem 1) and then used when needed, and this implies that time is saved when recommending.

Having discussed the nodes affected by the evidence, we must propagate this evidence through the network. Although general purpose algorithms do exist, they take exponential time with the number of parents when applied to a BN with the proposed topology [34]. Nevertheless, considering that the evidence only affects user nodes and the conditional independence statements represented in the network, the a posteriori probabilities for the collaborative nodes can be computed efficiently by using the advantages of the canonical weighted-sum representation in Eq. (8).

**Theorem 1.** *Let $l_{X_a}$ denote the number of states that $X_a$ takes in the collaborative BN network and let $Y_j$ be a node in $Pa(X_a)$. Let us assume that the set of conditional probability distributions over $X_a$ are expressed using a canonical weighting scheme, i.e.*

$$\Pr(x_{a,s}|pa(X_a)) = \sum_{Y_j \in Pa(X_a)} w(y_{j,t}, x_{a,s}),$$

*where $y_{j,t}$ is the value that variable $Y_j$ takes in the configuration $pa(X_a)$ and $w(\cdot,\cdot)$ are a set of non-negative weights verifying that*

$$\sum_{s=1}^{l_{X_a}} \sum_{Y_j \in Pa(X_a)} w(y_{j,t}, x_{a,s}) = 1, \quad \forall pa(X_a).$$

*If the evidence, $ev$, is only on ancestors of $X_a$, the exact a posteriori probabilities can then be computed with the following formula*:

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

It is interesting to note that when there is no evidence we can compute the a priori probabilities for all the nodes in the BN using this result. The proof of this theorem can be found in the appendix of this paper.

### 7.2. Vote recommending: searching for the proper fuzzy label

Once we have computed the a posteriori probability distribution $\Pr(V_a|ev)$, the problem is to decide on the final rating that the system might recommend to the users. Since the output must be a fuzzy label on $\mathcal{S}_L$ describing the user's

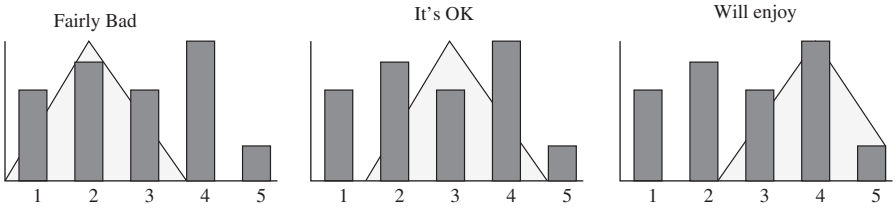


Fig. 4. Similarity between predicted ratings and fuzzy labels.

satisfaction degree, the problem is to look for the label $l$ which best matches the probability distribution $\Pr(V_a|ev)$. In this paper, we will consider two different alternatives:

PFE Computing probabilities of fuzzy events of the form 'Pr(rateis$l$)': these probabilities should be computed using Zadeh's definition of '*X is l*' [47]:

**Definition 1.** Let $X$ be a random variable with values in $\Omega$ and Pr be the probability distribution over $\Omega$ and let $l$ be a fuzzy subset of $\Omega$, then $\Pr(\text{Xisl}) = \sum_{x\in\Omega} \mu_l(x) \times \Pr(x)$.

Therefore, in our case, the rating is selected by computing the a posteriori probability for each fuzzy event $l$, $l \in \mathcal{S}_L$, and then returning the most probable fuzzy set. Using these probabilities, which are closer to the idea of expectation over $\mathcal{S}$, the final rate can be easily computed by means of

$$\text{vote} = \arg\max_l \left\{ \sum_{s=1}^{r} \mu_l(s) \Pr(V_a = s|ev) \right\}. \tag{14}$$

For example, considering the data in Table 2 and the set of triangular fuzzy labels described in Eq. (10), the system will recommend the vote $l_3 = OK$. It should be noted that if we consider the rates as crisp events, this alternative implies the selection of the vote with the highest a posteriori probability, i.e. vote $= \arg\max_s\{\Pr(V_a = s|ev)\}$ which in the example coincides with vote 4, i.e. *Will enjoy*.

SPF Use a similarity measure between the a posteriori probabilities and the fuzzy labels, giving as output the fuzzy label which is most similar to the a posteriori values. A direct similarity measure cannot be applied since we are talking not only about ambiguity in the description of a vote (fuzzy labels) but also about the undecidability on the predicted rating (probabilities). In order to achieve this objective, it is therefore necessary to make transformations so as to allow both fuzzy labels and probability values to be compared with a same language: the Possibility Theory language [48].

We will use $\Pi_{ev}(V_a)$ to denote the possibility distribution over the ratings obtained after transforming the a posteriori probability distribution $\Pr(V_a|ev)$ and $\Pi_l(V_a)$ to denote the possibility distribution representing the fuzzy label $l$. We can then use a similarity measure between them in order to select the best rate. Fig. 4 illustrates this idea, where on one side we show a possibility distribution on the values in $\mathcal{S}$ and on the other we have the fuzzy labels describing the rates, $l \in \mathcal{S}_L$. In this case, the system will return the label which best matches the (a posteriori) possibility values.

Many measures have been published to express the similarity or equality between two fuzzy sets [11,50]. In this paper, we propose to use a similarity measure based on a geometric distance model, the idea being that the smaller the distance between the possibility distributions $\Pi_A$ and $\Pi_B$, the greater the similarity between them. Thus, following Zwick et al. [50], given two possibility distributions $\Pi_A$ and $\Pi_B$ a one parameter class of distance functions can be defined:

$$d_z(\Pi_A, \Pi_B) = \left[ \sum_{x=1}^{n} abs(\pi_A(x) - \pi_B(x))^z \right]^{1/z}. \tag{15}$$

In this paper, we propose to use the parameter $z = 2$, and therefore the predicted vote will be the one with the lowest $d_2$ values, i.e.

$$\text{vote} = \arg\min_l \{d_2(\Pi_{ev}(V_a), \Pi_l(V_a))\}. \tag{16}$$

Table 3
Making decisions about the recommended vote

| $\mathcal{S}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\Pi_{ev}(V_a)$ | 0.666 | 0.833 | 0.666 | 1.0 | 0.166 |
| $\mathcal{S}_L$ | $l_1$ | $l_2$ | $l_3$ | $l_4$ | $l_5$ |
| $d_2(\Pi_{ev}, \Pi_l)$ | 1.301 | 1.054 | 0.971 | 1.130 | 1.589 |

We shall now look at how the above transformations could be carried out:

- $\Pr(V_a|ev) \longrightarrow \Pi_{ev}(V_a)$: On the basis that a numerical degree of possibility can be viewed as an upper bound of a probability degree [48], it is possible to change representation from probability to possibility, and vice versa. Changing a probability distribution into a possibility distribution entails loss of information as the variability expressed by a probability measure is changed into incomplete knowledge or imprecision. Many papers have been published which study the probability/possibility transformation problem [14,16,25]. These transformations produce possibility distributions verifying that for each event the possibility degree is always greater than or equal to its corresponding probability degree, i.e. $\forall A,\ \Pi(A) \geqslant \Pr(A)$. In this paper, we propose the use of the following transformation, [3] where in order to obtain a possibility measure from a probability distribution it is sufficient to normalize it by using the value of maximum probability [25], i.e.

$$\pi(x_i) = \frac{\Pr(x_i)}{\max_{j=1}^{n} \Pr(x_j)}. \tag{17}$$

- $l \in \mathcal{S}_L \longrightarrow \Pi_l(V_a)$: Given a fuzzy label representing that the 'rate is $l$', it is quite easy to obtain a possibility measure from it. We need only consider the induced possibility distribution $\Pi_l$ which equates the possibility of the rate taking the value $s \in \mathcal{S}$ to $\mu_l(s)$, i.e. $\pi_l(s) = \mu_l(s)$.

Continuing with the above example, in Table 3 we present $\Pi_{ev}$ and the distance values for each $l \in \mathcal{S}_L$, i.e. $d_2(\Pi_{ev}, \Pi_l)$. These values have been calculated using the description of the fuzzy labels in Eq. (10). In this case, the vote predicted is the label $l_3$, *OK*.

### 7.3. System efficiency

The presented model is quite efficient when trying to predict an active user's rating. In this case, since the model is learnt offline and the necessary a priori probability distributions can be computed and stored in a pre-processing step (offline), we need only focus online on the collaborative node representing the user. When this node is loaded into the memory, the computations can be performed in a time which is linear to the number of parents that a collaborative node has, $O(|Pa(V_a)|)$ (see Theorem 1). We can therefore conclude that the model might be used in real industrial applications, when the users are logged online to the system, even when we have a large database of ratings.

## 8. Empirical analysis

This section presents some experimental results about the performance of the system. We decided to use MovieLens for the following reasons: it is publicly available; it has been used in many collaborative-based RS models (see Section 8.3) and so we can compare our results with those obtained; and (as discussed throughout the paper) the system stores crisp values although vague concepts might be used when rating. For these reasons we believe that it is a good benchmark for our purposes. The MovieLens data set was collected by the GroupLens Research Project at the University of Minnesota during the seven-month period between September 19th, 1997 and April 22nd, 1998 with votes ranging from 1 to 5. The data set contains 1682 movies and there are 943 users in the data set, containing 100,000 transactions on the scale 1–5. In order to perform five-fold cross validation, we have used the data sets U1.base and U1.test through U5.base and U5.test provided by MovieLens which split the collection into 80% for training and 20% for testing, respectively.

[3] We have experimentally considered different transformations, but the best results have been obtained with this one.

Table 4
Accuracy of the baseline (probabilistic) model

| | Number of parents | | | | | | |
|---|---|---|---|---|---|---|---|
| | 5 | 10 | 20 | 30 | 50 | 75 | 100 |
| MAE | 0.8127 | 0.7918 | 0.7872 | 0.7861 | 0.7886 | 0.7938 | 0.7966 |
| SDev | 0.0059 | 0.0041 | 0.0037 | 0.0041 | 0.0051 | 0.0045 | 0.0047 |

In order to test the performance of our model, we will measure its capability to predict user's true rating or preferences, i.e. system accuracy. Following [19], we propose to use the mean absolute error (MAE) which measures how close the system predictions are to the user's rating for each movie by considering the average absolute deviation between a predicted rating and the user's true rating:

$$MAE = \frac{\sum_{i=1}^{N} abs(p_i - r_i)}{N} \tag{18}$$

with $N$ being the number of cases in the test set, $p_i$ the vote predicted for a movie, and $r_i$ the true rating.

There are certain research questions that we would like to evaluate in this section. Our experimentation [4] will be directed towards finding answers to these questions:

(1) *Is the use of vague concepts when recommending relevant*? If the answer is yes, *Which strategy is most effective*? In this case, we must answer questions such as *What is the optimal number of parents for a collaborative node? Which is the best fuzzy description*? *What is the effect of those users who did not see a movie*? etc.
(2) *How good is the model when the problem is to find good items*?
(3) *Is the proposed model competitive*?

In each result table, we will present the average MAE obtained after repeating the experiment with each training and test set and the standard deviation for the five experiments.

### 8.1. Is the use of vague concepts when recommending relevant?

Before answering this question, we must measure the accuracy of the system without considering fuzzy concepts, i.e. assuming crisp ratings. In this case, we have a purely probabilistic model, which will be considered as the baseline. Thus, for each active user, we will recommend the most probable rating, i.e. vote $= \arg\max_s \Pr(V_a = s|ev)$. Table 4 shows the MAE values obtained by our model when using different sizes for the parent sets at collaborative nodes.

Perhaps a preliminary conclusion from these results is that the number of parents is relevant to the accuracy of the system so that having too few or too many parents worsens system performance. Since the number of parents will be a variable used throughout the experimentation, and in order to reduce the number of values presented, in the rest of the paper we will consider only parent sets with a size of 10, 30, 50 and 75.

Focusing on our question, *Is the use of vague concepts when recommending relevant*? we must study three different situations: the first which considers Fuzzy observations as Input and Crisp criteria in the system Outputs (denoted by FICO); the second which considers Crisp Inputs and Fuzzy Outputs (CIFO); and the third which considers Fuzzy Inputs and Fuzzy Outputs (FIFO).

In order to perform these experiments, we must consider different sets of fuzzy labels, $FL_j$, $j = 0, \ldots, 3$ where each $FL_j$ represents the description of the set of linguistic variables associated with the vague ratings from *Awful* to *Must see*. In order to define these sets, we will use the notation $\mu_{\mathrm{FL}_j,l}(s)$ to represent the membership degree of the value $s$ to the label $l$ in the set of fuzzy labels $FL_j$. These values will be defined by means of a vector of membership degrees $FL_j = (\delta_0, \delta_1, \ldots, \delta_n)$ with $\delta_k \in [0, 1]$ using the following expression:

$$\mu_{\mathrm{FL}_j,l}(s) = \delta_i \quad \text{if } abs(l - s) = i.$$

[4] We should say that there are various parameters which have been fixed in preliminary experimentation. In order to focus on those questions that seem relevant for our purposes, we have decided not to present these experimental results here. We will only say that $\alpha = 0.2$, $\beta = 1$ and $q_i = \frac{1}{5}$.

Table 5
MAE and standard deviation metrics for the FICO model

| | Number of parents | | | |
|---|---|---|---|---|
| | 10 | 30 | 50 | 75 |
| $FL_0$ | $0.7931 \pm 0.0046$ | $0.7923 \pm 0.0050$ | $0.79586 \pm 0.0048$ | $0.8004 \pm 0.0047$ |
| $FL_1$ | $0.7967 \pm 0.0051$ | $0.7954 \pm 0.0047$ | $0.8000 \pm 0.0049$ | $0.8051 \pm 0.0048$ |
| $FL_2$ | $0.8049 \pm 0.0050$ | $0.8041 \pm 0.0055$ | $0.8092 \pm 0.0052$ | $0.8138 \pm 0.0050$ |
| $FL_3$ | $0.8178 \pm 0.0055$ | $0.8108 \pm 0.0037$ | $0.8143 \pm 0.0056$ | $0.8282 \pm 0.0051$ |

Table 6
MAE and standard deviation metrics for the CIFO model

| | | Number of parents | | | |
|---|---|---|---|---|---|
| | | 10 | 30 | 50 | 75 |
| $FL_0$ | PFE | $0.7542 \pm 0.0041$ | $0.7487 \pm 0.0039$ | $0.7503 \pm 0.0046$ | $0.7539 \pm 0.0044$ |
| | SFP | $0.7562 \pm 0.0044$ | $0.7506 \pm 0.0038$ | $0.7516 \pm 0.0046$ | $0.7549 \pm 0.0047$ |
| $FL_1$ | PFE | $0.7437 \pm 0.0052$ | $0.7433 \pm 0.0046$ | $0.7477 \pm 0.0049$ | $0.75271 \pm 0.0049$ |
| | SFP | $0.7433 \pm 0.0051$ | $0.7399 \pm 0.0045$ | $0.7431 \pm 0.0045$ | $0.7468 \pm 0.0047$ |
| $FL_2$ | PFE | $0.7529 \pm 0.0043$ | $0.7539 \pm 0.0034$ | $0.7610 \pm 0.0038$ | $0.7685 \pm 0.0042$ |
| | SFP | $0.7458 \pm 0.0047$ | $0.7426 \pm 0.0046$ | $0.74698 \pm 0.0040$ | $0.7520 \pm 0.0038$ |
| $FL_3$ | PFE | $0.7968 \pm 0.0043$ | $0.8130 \pm 0.0047$ | $0.8277 \pm 0.0053$ | $0.8412 \pm 0.0053$ |
| | SFP | $0.7681 \pm 0.0043$ | $0.7787 \pm 0.0042$ | $0.7894 \pm 0.0049$ | $0.8010 \pm 0.005$ |

Similarly, we use $FL_0 = (1, 0.25, 0, 0, 0)$, $FL_1 = (1, 0.5, 0, 0, 0)$ and $FL_2 = (1, 0.5, 0.25, 0, 0)$. For instance, the description of the set $FL_1$ coincides with the one presented in Eq. (10). Finally, $FL_3$ is a set of fuzzy labels sharped for the extreme ratings $l_1$ and $l_5$ (using the membership functions $\mu_{\mathrm{FL}_3,l_1} = \{1/1, 0.25/2, 0/3, 0/4, 0/5\}$ and $\mu_{\mathrm{FL}_3,l_5} = \{0/1, 0/2, 0/3, 0.25/4, 1/5\}$) and wider for medium ratings (using $\mu_{\mathrm{FL}_3,l_2} = \{0.5/1, 1/2, 0.5/3, 0/4, 0/5\}$ , $\mu_{\mathrm{FL}_3,l_4} = \{0/1, 0/2, 0.5/3, 1/4, 0.5/5\}$ and $\mu_{\mathrm{FL}_3,l_3} = \{0.25/1, 0.5/2, 1/3, 0.5/4, 0.25/5\}$).

*FICO model*: Table 5 shows the results of the FICO model. From these results, we can conclude that considering only the fuzzy interface at the inputs of the system does not outperform the performance of the baseline model. We believe that this is because each user has rated a large number of movies (MovieLens asks for at least 20 movies to begin the recommendations) and the probability distributions obtained from the crisp ratings are a better representation of the user's voting pattern than the one obtained by considering the rates as fuzzy observations.

*CIFO model*: We must consider the two proposed alternatives to select the user's recommended rating, i.e. to compute the probabilities of a fuzzy event (PFE) or use a similarity measure between probabilities and fuzzy values (SPF). Table 6 displays the results obtained in this experimentation. Various conclusions can be drawn from this table. The first conclusion is that the use of vague concepts in the system output considerably improves the system performance (comparing with the baseline there were improvements of the MAE metric to the order of 0.04). Secondly, the SPF criterion is in general preferable to the PFE, with better results in $FL_1$, $FL_2$ and $FL_3$ experiments and obtaining the best values when FL1 is used, and this implies a quite vague definition of the ratings. In addition, we obtain the same conclusions as in the previous experiments, and the system performance worsens when large parent sets are used.

*FIFO model*: Table 7 presents only the results obtained using the same label definition at the input and the output of the system.[5] We therefore found that taking fuzzy inputs into account worsens the behavior of the system. It should be noted that when FIFO is used, it is better for there to be a small number of parents, and we believe that in this case, the use of fuzzy inputs helps to describe the user's behavior.

[5] Different combinations have also been tested but they do not improve the presented results nor do they invalidate the conclusions obtained.

Table 7
MAE and standard deviation metrics for the FIFO model

| | | Number of parents | | | |
|---|---|---|---|---|---|
| | | 10 | 30 | 50 | 75 |
| $FL_0$ | PFE | $0.7527 \pm 0.0044$ | $0.7517 \pm 0.0040$ | $0.7539 \pm 0.0039$ | $0.7581 \pm 0.0035$ |
| | SFP | $0.7541 \pm 0.0044$ | $0.7530 \pm 0.0038$ | $0.7570 \pm 0.0041$ | $0.7602 \pm 0.0037$ |
| $FL_1$ | PFE | $0.7478 \pm 0.0055$ | $0.7517 \pm 0.0052$ | $0.7568 \pm 0.0052$ | $0.7618 \pm 0.0049$ |
| | SFP | $0.7450 \pm 0.0054$ | $0.7477 \pm 0.0048$ | $0.7518 \pm 0.0050$ | $0.7563 \pm 0.0048$ |
| $FL_2$ | PFE | $0.7708 \pm 0.0044$ | $0.7763 \pm 0.0037$ | $0.7842 \pm 0.0043$ | $0.7929 \pm 0.0048$ |
| | SFP | $0.7586 \pm 0.0039$ | $0.7605 \pm 0.00379$ | $0.7655 \pm 0.0038$ | $0.7714 \pm 0.0045$ |
| $FL_3$ | PFE | $0.8259 \pm 0.0041$ | $0.8475 \pm 0.0050$ | $0.8606 \pm 0.0052$ | $0.8741 \pm 0.0051$ |
| | SFP | $0.7940 \pm 0.0043$ | $0.8079 \pm 0.0046$ | $0.8202 \pm 0.0048$ | $0.8327 \pm 0.0051$ |

By way of answer to this question, we can therefore conclude:

(1) It is preferable to use vague concepts when recommending. There is a clear ordering between the models: CIFO $\prec$ FIFO $\prec$ Baseline $\prec$ FICO. From this conclusion, we have that in the case of having a purely fuzzy rating scheme in the inputs, it might be convenient to study the benefits of considering these ratings as crisp values and in the case of a crisp rating scheme (such as MovieLens), it might be convenient to use fuzzy definitions in the output interface.
(2) With respect to the membership functions used to describe the vague ratings, it is better to use sharp fuzzy labels.
(3) The best accuracy is obtained when using a fuzzy similarity measure (SFP) to compare the a posteriori probability values in the collaborative node with the set of vague ratings.
(4) It is preferable to have medium-sized parent sets at collaborative nodes: a small number of parents seems to be insufficient because it reduces the possibility that some of them have seen the movie, whereas a large number of parents might introduce the noise of those users with a low similarity with the active user.

There is another question that we want to study: in order to compute the probability values used to predict the rating, we use information of *all the users related* to the active collaborative node $V_a$. We therefore ask ourselves *What is the effect on the recommendations of those* (*similar*) *users who did not rate a movie*?

More specifically, using the results in Theorem 1 we have that

$$\Pr(V_a = s|ev) = \sum_{U_j \in Pa^+(V_a)} w(u_{j,t}, v_{a,s}) + \sum_{U_i \in Pa^-(V_a)} \sum_{k=1}^{l_{Y_i}} w(u_{i,k}, v_{a,s}) \cdot \Pr(u_{i,k}) \tag{19}$$

with $Pa^+(V_a)$ and $Pa^-(V_a)$ being the set of parents who have and have not seen the movie, respectively, and $t$ is how user $U_j \in Pa^+(V_a)$ has rated this movie. The problem that we are now tackling is the effect of those users on $Pa^-(V_a)$. It should be noted that this question is also related to the number of parents that a collaborative node has: the lower the number of parents, the greater the possibility of $Pa^+(V_a)$ being empty, whereas a larger number of parents will mean that there are more elements in $Pa^-(V_a)$. We will study both situations separately.

(1) $Pa^+(V_a) = \emptyset$. Looking at Table 8, which displays the mean number of times (and the standard deviation) that $Pa^+(V_a)$ is empty we found that there are many situations where the recommendations have been made without information, i.e. none of the related users saw the movie. In this case, we believe that it might be more convenient to use the a priori probabilities of the active user (his/her own rating pattern) to produce the recommendations, without considering the information given by its parents. Thus, if $Pa^+(V_a) = \emptyset$ then we will use $\Pr(U_a)$ (instead of $\Pr(V_a|ev)$) to recommend. We will only present the results obtained with the CIFO model (see Table 9). Comparing these results with those presented in Table 6, we can conclude that there is an improvement in system accuracy in

Table 8
Mean number of times that $Pa^+(V_a)$ is empty

| Number of parents | | | |
|---|---|---|---|
| 10 | 30 | 50 | 75 |
| $1283.2 \pm 31.29$ | $392.4 \pm 12.54$ | $242.2 \pm 9.19$ | $176.8 \pm 7.9$ |

Table 9
MAE and standard deviation metrics for the CIFO model using the a priori user pattern of rating when $Pa^+(V_a)$ is empty

| | | Number of parents | | | |
|---|---|---|---|---|---|
| | | 10 | 30 | 50 | 75 |
| $FL_0$ | PFE | $0.7416 \pm 0.0037$ | $0.7463 \pm 0.0037$ | $0.7483 \pm 0.0044$ | $0.7514 \pm 0.0042$ |
| | SFP | $0.7540 \pm 0.0041$ | $0.7483 \pm 0.0036$ | $0.7496 \pm 0.0044$ | $0.7534 \pm 0.0047$ |
| $FL_1$ | PFE | $0.7395 \pm 0.0046$ | $0.7404 \pm 0.0042$ | $0.7456 \pm 0.0040$ | $0.7510 \pm 0.0047$ |
| | SFP | $0.7402 \pm 0.0044$ | $0.7357 \pm 0.00415$ | $0.7412 \pm 0.0042$ | $0.7452 \pm 0.00451$ |
| $FL_2$ | PFE | $0.7482 \pm 0.0033$ | $0.7502 \pm 0.00268$ | $0.7578 \pm 0.0032$ | $0.7657 \pm 0.0038$ |
| | SFP | $0.7432 \pm 0.0037$ | $0.7395 \pm 0.0039$ | $0.7442 \pm 0.0034$ | $0.7496 \pm 0.0035$ |
| $FL_3$ | PFE | $0.7893 \pm 0.0031$ | $0.8086 \pm 0.0039$ | $0.8240 \pm 0.0047$ | $0.8381 \pm 0.0048$ |
| | SFP | $0.7624 \pm 0.0032$ | $0.7748 \pm 0.00345$ | $0.7863 \pm 0.0044$ | $0.7982 \pm 0.0042$ |

all the situations. [6] It is also interesting to note that when using PFE, we obtain the best results when using smaller parent set sizes. With the SPF, the best results have generally been obtained with a parent set of size 30.

(2) Large $Pa^-(V_a)$. In this case, we will study what happens if we only consider those users who have rated the movie and for this, we will assume that $Pa^-(V_a) = \emptyset$ in some way. The idea is to consider how the new evidence has been spread through the model trying to reduce the bias that, a priori, the user should have for a given vote. We will therefore only measure the new piece of evidence that each candidate rating receives by means of the users who previously rated the movie. This new piece of evidence can be easily defined by considering the difference between the a priori (without evidence) and the a posteriori probability values, i.e. $\Pr(V_a = s|ev) - \Pr(V_a = s)$. With this idea, a new a posteriori probability for the active node is defined by means of

$$\Pr_d(v_{a,s}|ev) = \begin{cases} \dfrac{\Pr(v_{a,s}|ev) - \Pr(v_{a,s})}{\sum_{t^+} \Pr(v_{a,t^+}|ev) - \Pr(v_{a,t^+})} & \text{if } \Pr(v_{a,s}|ev) - \Pr(v_{a,s}) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

with $t^+$ being those ratings such that $\Pr(v_{a,t^+}|ev) - \Pr(v_{a,t^+}) > 0$.

In Table 10, we only present the results obtained using the CIFO model with the SPF criterion to select the rating and considering the active user's information when none of its parents have observed the item. This table also displays the results obtained with the new baseline (probabilistic) model returning the rating $s$ with the greatest value of $\Pr_d(v_{a,s}|ev)$. From this table, we can conclude that:

1. The best results are obtained with the baseline model, i.e. it is not interesting to use fuzzy outputs.
2. A large number of parents helps improve system performance. We believe that this is because we have more information when recommending. It should be noted that in the case of having weakly related parents, these do not introduce much noise since this fact has been taken into account in the weighting scheme.
3. Comparing the results with those in Table 9, we can also conclude that the use of the users in $Pa^-(V_a)$ (as in Eq. (19)) helps improve system accuracy.

[6] Similar performance has been obtained with the rest of the models.

Table 10
MAE and standard deviation metrics for the CIFO model using $\Pr_d(V_a|ev)$

| | | Number of parents | | | |
|---|---|---|---|---|---|
| | | 10 | 30 | 50 | 75 |
| Baseline | | $0.7898 \pm 0.0035$ | $0.7771 \pm 0.0037$ | $0.7745 \pm 0.0043$ | $0.7837 \pm 0.0040$ |
| $FL_0$ | SFP | $0.8514 \pm 0.0018$ | $0.7905 \pm 0.0041$ | $0.7815 \pm 0.0045$ | $0.7806 \pm 0.0044$ |
| $FL_1$ | SFP | $0.8587 \pm 0.0023$ | $0.7960 \pm 0.0051$ | $0.7891 \pm 0.0054$ | $0.7874 \pm 0.0057$ |
| $FL_2$ | SFP | $0.8983 \pm 0.0023$ | $0.8415 \pm 0.0053$ | $0.8355 \pm 0.0053$ | $0.7994 \pm 0.0045$ |
| $FL_3$ | SFP | $0.8648 \pm 0.0020$ | $0.8073 \pm 0.0050$ | $0.8355 \pm 0.0053$ | $0.8364 \pm 0.0050$ |

### 8.2. How good is the model when the problem involves finding good items?

When considering tasks such as *find good items*, the MAE metric might not be appropriate mainly because ranking results are returned to the user, who only views items at the top of the ranking. We must therefore consider a different accuracy metric considering the frequency with which a RS makes correct or incorrect decisions (classifications) about whether an item is good or not. Following [19], two different measures might be considered: *Recall*, which is defined as the ratio of relevant items selected, $N_{\mathrm{rs}}$, to the total number of relevant items available, $N_{\mathrm{r}}$, and *Precision*, which is defined as the ratio of relevant items selected, $N_{\mathrm{rs}}$, to the number of items selected, $N_{\mathrm{s}}$.

$$P = \frac{N_{\mathrm{rs}}}{N_{\mathrm{s}}}, \quad R = \frac{N_{\mathrm{rs}}}{N_{\mathrm{r}}}. \tag{21}$$

It is well known that these measures are inversely correlated and depend on the number of items returned. We will use the F1 metric [19] which is one of the most common metrics for combining $P$ and $R$ into a single value:

$$F1 = \frac{2PR}{P + R}. \tag{22}$$

This metric takes its values in [0, 1] verifying that the closer a value is to 1, the better the quality of the model when recommending good items.

When using these metrics, it is necessary for the items to be separated into two classes: *relevant* and *not relevant*. Since our rating scale is not binary, we need to transform it into a binary scale. Following [13], we propose that every movie rated with 4 or 5 be considered as relevant and all those rating from 1 to 3 to be not relevant. We also consider that all the items are retrieved.

Our objective is to study the ability of our system to find good items, although it has not been specifically designed for this purpose. For this reason, we will only present the final conclusions and not a detailed experimentation here.

(1) The performance of the model for this purpose is quite good, all the F1 values are in the interval from 0.7253 to 0.7534.
(2) We have systematically obtained the best results by combining $\Pr_d(V_a|ev)$ in the CIFO model of (Eq. (20)), the SFP criterion using $FL_1$ or $FL_2$ and large parent sets (50 or 75). The F1 values in these cases are in the interval from 0.7525 to 0.7534.

### 8.3. Is the proposed model competitive?

In order to answer this question, we must compare the performance of our model with other published RS. In order to perform this comparison, we should take into account that the selected collaborative models use the same data collection, i.e. MovieLens, and the same performance measure, i.e. MAE. Following [19], it can be found that when the best algorithms are " tuned to its optimum, they all produce similar measures of quality. We—and others—have speculated that we may reach some 'magic barrier' where natural variability may prevent us from getting much more accurate." This barrier is around the values 0.72–0.73 of the MAE metric [19] (the particular MAE values are: Li and Kim [26] got 0.735, Sarwar et al. [39,40] got 0.72, Chen and Yin [12] got 0.732 and Mobasher et al. [30] got 0.73) although there is one work which has passed this barrier [24] using *principal component analysis*,

obtaining the value of 0.70 for the MAE metric. With these data to hand, we can conclude that our model, obtaining a better MAE of 0.7357, is competitive with the best published standards. It is worth mentioning that when we do not consider those recommendations made without any information, i.e. when $Pa^+(V_a) = \emptyset$, a MAE result of 0.7247 is obtained.

## 9. Conclusions

In this paper, we have proposed a general Soft Computing-based model for collaborative recommendation. We have also studied the possibility of considering the set of ratings as vague concepts. Schematically, our system consists of three components: the first maps the input fuzzy rating to a probability distribution; the second uses probabilistic reasoning to compute the probability distribution over the expected vote; and the third computes the user's vote (the fuzzy set), thereby better representing this probability distribution. We are therefore using a fuzzy interface to enable the users to communicate with the system and this results in a considerable improvement in terms of system performance.

Taking into account efficiency considerations, the posterior probabilities for recommending are computed using a very efficient method based on canonical models. Guidelines of how to estimate the probability values from a (vague) data set and also how the RS interacts with the users have been given. It must be noted that the proposed model is quite general, since it can be applied to solve different recommendation tasks (such as *finding good items* or *predicting rates*). Throughout the process, computational aspects of the RS have been considered, such as the sparseness of the data and the fact that the ranking should be computed in real time.

By way of future work, we are planning to study mechanisms to incorporate better specifications of the products into the system and new methods for estimating the weights stored in the nodes of the BN. Nevertheless, we wonder, like [19], whether 'users are sensitive to a change in the mean absolute error of 0.01?' This observation suggests that we might explore different directions instead of merely continuing to improve the MAE metric. In the future, we therefore plan to study problems such as how our system can communicate its reasoning to the users, the minimum amount of data (ratings) required for us to yield accurate recommendations, or how to include item information when recommending.

## Appendix

The next theorem shows how it can be computed efficiently (in a polynomial time with the number of parents) the a posteriori probability of a variable when considering a canonical weighting scheme for a set of multivaluated variables.

**Theorem 1.** *Let $l_{X_a}$ denote the number of states that $X_a$ takes in the collaborative BN network and let $Y_j$ be a node in $Pa(X_a)$. Let us assume that the set of conditional probability distributions over $X_a$ are expressed using a canonical weighting scheme, i.e.*

$$\Pr(x_{a,s}|pa(X_a)) = \sum_{Y_j \in Pa(X_a)} w(y_{j,t}, x_{a,s}),$$

*where $y_{j,t}$ is the value that variable $Y_j$ takes in the configuration $pa(X_a)$ and $w(\cdot,\cdot)$ are a set of non-negative weights verifying that*

$$\sum_{s=1}^{l_{X_a}} \sum_{Y_j \in Pa(X_a)} w(y_{j,k}, x_{a,s}) = 1, \quad \forall pa(X_a).$$

*Then, if the evidence, ev, is only on ancestors of $X_a$, the exact a posteriori probabilities can be computed using the following formula:*

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

**Proof.** We know that

$$\Pr(x_{a,s}|ev) = \sum_{pa(X_i)} \Pr(x_{a,s}|pa(X_i), ev) \cdot \Pr(pa(X_i)|ev), \tag{23}$$

where the sum is taken over all the possible configurations of the set of variables $Pa(X_i)$. Since given the set of parents of $X_a$, $Pa(X_a)$, $X_a$ is independent of the evidence then

$$\Pr(x_{a,s}|ev) = \sum_{pa(X_i)} \Pr(x_{a,s}|pa(X_i)) \cdot \Pr(pa(X_i)|ev).$$

Substituting in the previous expression the value of $\Pr(x_{a,s}|pa(X_i))$ in Eq. (8), we obtain

$$\Pr(x_{a,s}|ev) = \sum_{pa(X_i)} \left( \sum_{Y_j \in Pa(X_a)} w(y_{j,t}, x_{a,s}) \right) \cdot \Pr(pa(X_i)|ev) \tag{24}$$

with $y_{j,t}$ being the value that variable $Y_j$ takes in the configuration $pa(X_a)$. Without losing generality, we will assume an order among the variables in $Pa(X_a)$, i.e. $Pa(X_a) = \{Y_1, Y_2, \ldots, Y_m\}$. Thus, the number of states of variable $Y_m$ is represented as $l_{Y_m}$. The previous expression can then be broken down into $l_{Y_m}$ parts, each including those configurations where the variable $Y_m$ takes the $k$th-value, $1 \leqslant k \leqslant l_{Y_m}$. In order to make this fact explicit, we will denote with $Pa^*(X_a)$ the first $m-1$ parents of $X_a$, i.e. $Pa^*(X_a) = \{Y_1, \ldots, Y_{m-1}\}$, $pa^*(X_a)$ also denotes a configuration of $Pa^*(X_a)$ and $\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle$ represents a configuration of $Pa(X_a)$ where variable $Y_m$ takes the $k$th-value.

$$\Pr(x_{a,s}|ev) = \sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_i), \mathbf{y_{m,k}} \rangle} \left( \sum_{Y_j \in Pa(X_a)} w(y_{j,t}, x_{a,s}) \right) \cdot \Pr(pa(X_i)|ev) \tag{25}$$

which is equal to

$$\sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle} \left( \sum_{Y_j \in Pa^*(X_a)} \left( w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev) \right) + w(\mathbf{y_{m,k}}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev) \right)$$

and grouping terms

$$\begin{aligned} \Pr(x_{a,s}|ev) = & \sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle} \sum_{Y_j \in Pa^*(X_a)} w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev) \\ & + \sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_i), \mathbf{y_{m,k}} \rangle} w(\mathbf{y_{m,k}}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev). \end{aligned} \tag{26}$$

Focusing on the second addend of this equation, since for a fixed value $\mathbf{y_{m,k}}$, $1 \leqslant k \leqslant m$, of variable $Y_m$ it is verified that

$$\sum_{\langle pa^*(X_a), \mathbf{y_{m,k}} > } w(\mathbf{y_{m,k}}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev)$$

is equal to

$$w(\mathbf{y_{m,k}}, x_{a,s}) \sum_{\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle} \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}} \rangle|ev)$$

which implies that we are summing over all the possible configurations in $pa^*(X_a)$, and it is therefore equivalent to

$$w(\mathbf{y_{m,k}}, x_{a,s}) \cdot \Pr(\mathbf{y_{m,k}}|ev).$$

Consequently, Eq. (26) becomes

$$\Pr(x_{a,s}|ev) = \sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_a),\mathbf{y_{m,k}}\rangle} \sum_{Y_j \in Pa^*(X_a)} w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}}\rangle|ev) + \sum_{k=1}^{l_{Y_m}} w(\mathbf{y_{m,k}}, x_{a,s}) \cdot \Pr(\mathbf{y_{m,k}}|ev). \quad (27)$$

We can now focus on the first addend in the previous expression, i.e.

$$\sum_{k=1}^{l_{Y_m}} \sum_{\langle pa^*(X_a),\mathbf{y_{m,k}}\rangle} \sum_{Y_j \in Pa^*(X_a)} w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}}\rangle|ev).$$

We can see how, independently of the value of variable $Y_m$, each sum in the configuration has $\sum_{Y_j \in Pa^*(X_a)} w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}}\rangle|ev)$ in common. Therefore, this expression is equivalent to

$$\sum_{pa^*(X_a)} \sum_{Y_j \in Pa^*(X_a)} \sum_{k=1}^{l_{Y_m}} w(y_{j,t}, x_{a,s}) \cdot \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}}\rangle|ev)$$

which is equal to

$$\sum_{pa^*(X_a)} \sum_{Y_j \in Pa^*(X_a)} w(y_{j,t}, x_{a,s}) \sum_{k=1}^{l_{Y_m}} \Pr(\langle pa^*(X_a), \mathbf{y_{m,k}}\rangle|ev).$$

Therefore, we could unify these $l_{Y_m}$ last addends into a single one since for a fixed configuration $pa^*(X_a)$ we are summing over all the possible values taken by variable $Y_m$, which represents a marginalization operation over the variable $Y_m$ and consequently the variable $Y_m$ will be removed from the resultant addend, i.e. the above expression is equivalent to

$$\sum_{pa^*(X_i)} \sum_{Y_j \in Pa^*(X_i)} w(y_{j,t}, x_{a,s}) \cdot \Pr(pa^*(x_{a,s})|ev).$$

Therefore, the a posteriori probability becomes

$$\Pr(x_{a,s}|ev) = \sum_{pa^*(X_i)} \sum_{Y_j \in Pa^*(X_i)} w(y_{j,t}, x_{a,s}) \cdot \Pr(pa^*(x_{a,s})|ev) + \sum_{k=1}^{l_{Y_m}} w(\mathbf{y_{m,k}}, xa, s) \cdot \Pr(\mathbf{y_{m,k}}|ev).$$

It should be noted that the first addend is completely analogous to the initial expression, Eq. (24), but where the variable $Y_m$ has been removed. We now repeat the process applied to this first addend to remove a new variable $Y_{m-1}$ and extract the addends $\sum_{k=1}^{l_{Y_{m-1}}} w(\mathbf{y_{m-1,k}}, x_{a,s}) \cdot \Pr(\mathbf{y_{m-1,k}}|ev)$. By repeating the process until we have removed all the terms, we obtain a final expression of the probability of $X_i$ given the evidences:

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

## References

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

[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, NY, USA, 1981.

[3] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: 14th Conf. on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
[4] C. Butz, Exploiting contextual independencies in web search and user profiling, in: Proc. of World Congress on Computational Intelligence, 2002, pp. 1051–1056.
[5] L.M. de Campos, J.M. Fernández-Luna, M. Gómez, J.F. Huete, A decision-based approach for recommending in hierarchical domains, Lecture Notes in Computer Science 3571 (2005) 123–135.
[6] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, Generalizing e-bay.net: an approach to recommendation based on probabilistic computing, in: 1st Workshop on Web Personalization, Recommender Systems and Intelligent User Interface, 2005, pp. 24–33.
[7] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, A Bayesian network approach to hybrid recommending systems., in: Eleventh Internat. Conf. of Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2006, pp. 2158–2165.
[8] Y. Cao, Y. Li, X. Liao, Applying fuzzy logic to recommend consumer electronics, in: ICDCIT, 2005, pp. 278–289.
[9] J. Carbo, J. Molina, Agent-based collaborative filtering based on fuzzy recommendations, Int. J. Web Eng. Technol. 1 (4) (2004) 414–426.
[10] R. Chau, C.-H. Yeh, Fuzzy multilingual information filtering, in: IEEE Internat. Conf. on Fuzzy Systems, 2003, pp. 767–781.
[11] S.-M. Chen, M.-S. Yeh, P.-Y. Hsiao, A comparison of similarity measures of fuzzy values, Fuzzy Sets and Systems 72 (1995) 79–89.
[12] J. Chen, J. Yin, Recommendation based on influence sets, in: Proceedings of the Workshop on Web Mining and Web Usage Analysis, 2006.
[13] B. Dahlen, J. Konstan, J. Herlocker, N. Borchers, J. Rield, Jump-starting movielens: user benefits of starting a collaborative filtering system "dead data", in: TR98-017 University of Minnesota, 1998.
[14] M. Delgado, S. Moral, On the concept of possibility-probability consistence, Fuzzy Sets and Systems 21 (3) (1987) 311–318.
[15] D. Dubois, E. Hüllermeier, H. Prade, Fuzzy methods for case-based recommendation and decision support, J. Intell. Inform. Systems 27 (2) (2006) 95–115.
[16] D. Dubois, H. Prade, S. Sandri, On possibility/probability transformations, in: 4th Internat. Fuzzy Systems Association (IFSA'91) Congress, Vol. Mathematics, 1991, pp. 50–53.
[17] A. Gonzalez, A learning methodology in uncertain and imprecise environments, Internat. J. Intell. Systems 10 (1995) 357–371.
[18] D. Heckerman, D.M. Chickering, C. Meek, R. Rounthwaite, C. Kadie, Dependency networks for inference, collaborative filtering, and data visualization,, J. Mach. Learn. Res. 1 (2001) 49–75.
[19] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.
[20] T. Hofmann, J. Puzicha, Latent class models for collaborative filtering, in: 16th Internat. Joint Conf. on Artificial Intelligence, 1999, pp. 688–693.
[21] J.Y. Junzhon Li, C. Liu, N. Zhong, Bayesian networks structure learning and its application to personalized recommendation in a B2C portal, IEEE/WIC/ACM Internat. Conf. of Web Intelligence, 2004.
[22] S. Kangas, Collaborative filtering and recommendation systems, in: VTT Information Technology, 2002.
[23] J. Kim, E. Lee, Xfc—xml based on fuzzy clustering—method for personalized user profile based on recommendation system, in: IEEE Conf. on Cybernetics and Intelligent Systems, 2004, pp. 1202–1206.
[24] D. Kim, B. Yum, Collaborative filtering based on iterative principal component analysis, Expert Systems Appl. 28 (4) (2005) 823–830.
[25] G. Klir, B. Parviz, Probability-possibility transformations: a comparison, Internat. J. General Systems 21 (1992) 291–310.
[26] Q. Li, B. Kim, Clustering approach for hybrid recommender system, in: IEEE/WIC Proc. Internat. Conf. on Web Intelligence, 2003, pp. 33–38.
[27] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing 7 (1) (2003) 76–80.
[28] B. Miller, I. Albert, S. Lam, J. Konstan, J. Rield, Movielens unplugged: experiences with an occasionally connected recommender systems, in: Proc. Internat. Conf. Intelligent User Interfaces, 2002, pp. 263–266.
[29] K. Miyahara, M.J. Pazzani, Collaborative filtering with the simple Bayesian classifier, in: Pacific Rim Internat. Conf. on Artificial Intelligence, 2000, pp. 679–689.
[30] B. Mobasher, Y. Z. X. Jin, Semantically Enhanced Collaborative Filtering on the Web, Lecture Notes on Artificial Intelligence, Vol. 3209, 2004, pp. 57–76.
[31] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: DL '00: Proc. Fifth ACM Conf. on Digital Libraries, ACM Press, New York, NY, USA, 2000, pp. 195–204.
[32] D. O'Sullivan, B. Smyth, D. Wilson, K. McDonald, A. Smeaton, Improving the quality of personalized electronic program guide, User modeling User-Adapted Interact. 14 (2004) 5–35.
[33] M. Pazzani, D. Billsus, Learning and revising user profiles: the identification of interesting, Mach. Learning 27 (1997) 313–331.
[34] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
[35] A. Popescu, L. Ungar, D. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, in: 17th Conf. on Uncertainty in Artificial Intelligence, Seattle, Washington, 2001, pp. 437–444.
[36] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (3) (1997) 56–58.
[37] V. Robles, P. Larrañaga, J. Peña, O. Marbán, J. Crespo, M. Pérez, Collaborative Filtering Using Interval Estimation Naive Bayes, in: Lecture Notes in Computer Science, 2003, pp. 46–53.
[38] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1983.
[39] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of dimensionality reduction in recommender system—a case study, in: Proc. ACM WebKDD, 2000.
[40] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proc. ACM World Wide Web Conf., 2001, pp. 285–295.
[41] S.N. Schiaffino, A. Amandi, User profiling with case-based reasoning and bayesian networks, in: IBERAMIA-SBIA 2000 Open Discussion Track, 2000, pp. 12–21.

[42] H. Stormer, N. Werro, D. Risch, Recommending Products by the mean of a Fuzzy Classification, in: Proc. European Conf. on Collaborative Electronic Commerce Technology and Research, 2006.
[43] H. Toth, Probabilities and fuzzy events: an operational approach, Fuzzy Sets and Systems 48 (1) (1992) 113–127.
[44] R.R. Yager, Probabilities from fuzzy observations, Inform. Sci. 32 (1984) 1–31.
[45] R.R. Yager, Fuzzy logic methods in recommender systems, Fuzzy Sets and Systems 136 (2) (2003) 133–149.
[46] Y. Yoshinari, W. Pedrycz, K. Hirota, Construction of fuzzy models through clustering techniques, Fuzzy Sets and Systems 54 (2) (1993) 157–165.
[47] L.A. Zadeh, Probability measures from fuzzy events, Math. Anal. Appl. 23 (1968) 421–427.
[48] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.
[49] P. Zigoris, Y. Zang, Bayesian adaptive user profiling with explicit and implicit feedback, in: Conf. on Information and Knowledge Management, 2006, pp. 397–404.
[50] R. Zwick, E. Carlstein, D. Budescu, Measures of similarity among fuzzy concepts: a comparative analysis, lnternat. J. Approx. Reason. 1 (1987) 221–242.