International Journal of Approximate Reasoning 51 (2010) 785–799

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks ☆

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

Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación, CITIC-UGR Universidad de Granada, C.P, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 15 February 2008
Received in revised form 5 April 2010
Accepted 5 April 2010
Available online 11 April 2010

Keywords:
Content-based filtering
Collaborative filtering
Hybrid recommender systems
Bayesian networks
MovieLens
IMDB

ABSTRACT

Recommender systems enable users to access products or articles that they would otherwise not be aware of due to the wealth of information to be found on the Internet. The two traditional recommendation techniques are content-based and collaborative filtering. While both methods have their advantages, they also have certain disadvantages, some of which can be solved by combining both techniques to improve the quality of the recommendation. The resulting system is known as a hybrid recommender system.

In the context of artificial intelligence, Bayesian networks have been widely and successfully applied to problems with a high level of uncertainty. The field of recommendation represents a very interesting testing ground to put these probabilistic tools into practice.

This paper therefore presents a new Bayesian network model to deal with the problem of hybrid recommendation by combining content-based and collaborative features. It has been tailored to the problem in hand and is equipped with a flexible topology and efficient mechanisms to estimate the required probability distributions so that probabilistic inference may be performed. The effectiveness of the model is demonstrated using the MovieLens and IMDB data sets.

1. Introduction

Recommender systems (RSs) attempt to discover user preferences, and to learn about them in order to anticipate their needs. Broadly speaking, a recommender system provides specific suggestions about items (products or actions) within a given domain, which may be considered of interest to the given active user [1]. Formally, in a hybrid recommending framework, there exists a large number m of items or products $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$, which are described by a set of l attributes or features, $\mathcal{F} = \{F_1, F_2, \ldots, F_l\}$, and each product is specified by one or several. There is also 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 $\mathcal{I}$. Under this formulation we distinguish two different problems:

- Given an item not rated, predicting the rating that the user would give.
- Given a user, find the best items and their ratings for being recommended, showing the results ordered by predicted rating.

☆ Work jointly supported by the Spanish Ministerio de Educación y Ciencia (TIN2008-06566-C04-01) and the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

* Corresponding author.
E-mail addresses: lci@decsai.ugr.es (L.M. de Campos), jmfluna@decsai.ugr.es (J.M. Fernández-Luna), jhg@decsai.ugr.es (J.F. Huete), mrueda@decsai.ugr.es (M.A. Rueda-Morales).

0888-613X/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2010.04.001

Although both notions are closely related, this paper deals with the first type, i.e. rating prediction. The usual formulation of the problem is then to predict how an active user might rate an unseen item.

Many different approaches to the recommender system problem have been published [2–4], using methods from machine learning, approximation theory, and various heuristics. Independently of the technique used and based on how the recommendations are made, recommender systems are usually classified [3] into the following categories: *Collaborative filtering systems* that attempt to identify groups of people with similar tastes to those of the user and recommend items that they have liked and *Content-based recommender systems* which use content information in order to recommend items similar to those previously preferred by the user.

Generally, collaborative systems report a better performance than content-based approaches, but their success relies on the presence of a sufficient number of user ratings [3,5,6,4,7]. Such systems have the drawback that they suffer from the item cold-start problems which occur when recommendations must be made on the basis of few recorded ratings [8,3]. These problems arise because the similarity analysis is not accurate enough. In these situations the use of a content-based approach appears as an alternative. Nevertheless, this approach has its own limitations. For example, the keywords used to represent the content of the items might not be very representative. Also, content-based approaches suffer the limitation of making accurate recommendations to users with very few ratings.

A common approach to solve the problems of the above techniques is to combine both content-based and collaborative information into a *hybrid recommender system* [9]. Different hybridization methods [3,6,9,10] have been proposed, such as the use of weighted criterion (the scores of different recommendation components are combined numerically), the use of a switching mechanism (the system chooses among recommendation components and applies the selected one) or even the presentation of the two recommendations together, leaving the decision in the user's hands. Nevertheless, a common problem with these methods is that the parameters controlling the hybridization have to be tuned.

This is the setting for the proposal presented in this paper, i.e. the design of a hybrid system with the aim of predicting how an active user should rate a given item. Particularly, we will explore the use of Bayesian network formalism to represent the relationships among users $\mathcal{U}$, items $\mathcal{I}$ and features $\mathcal{F}$, the elements involved in the recommendation processes. By using Bayesian networks, we can 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.

In our proposal we shall distinguish two different parts: The first one is used to represent the knowledge that we have about how the active user rates the items, i.e. the user profile, which includes both content-based and collaborative information. The second component represents those relationships related to the target item. We would like to say that content-based information is not only used to improve the active user knowledge, but also this information has been used to improve the knowledge at the collaborative level. This is possible because we have a hybrid model where all the components are represented under the same formalism. By means of this fact we can explore the importance of the different elements in the quality of the predictions.

In order to present the model, this paper is structured in the following way. The following section introduces recommender systems and reviews the related work. Section 3 describes the model from a topological point of view. How to use the recommender model and how inference is performed efficiently are explained in Section 4. Section 5 discusses the probability distribution estimation. In order to determine the performance of the proposed model, it is evaluated in Section 6. Finally, Section 7 presents our conclusions and outlines future lines of research.

2. Related work and preliminaries

Based on how the recommendations are made, recommender systems are classified into:

- *Content-based recommender systems* that [3] store content information about each item to be recommended. This information will be used to recommend items similar to those previously preferred by the user, 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). Focusing on probabilistic approaches, learning as a constraint satisfaction problem is considered in [11], where the user profile is learnt by considering contextual independence. By assuming independence between variables, Bayesian classifiers have also been used in [12,13] to estimate the probability of an item belonging to a certain class (relevant or irrelevant) given the item description. Also, Bayesian networks [14,15] have been used to model the item's description.
- *Collaborative filtering systems* [3] attempt to identify groups of people with similar tastes to those of the user and recommend items that they have liked. According to [16], collaborative recommender systems can be grouped into *memory-based* and *model-based* approaches.

On the one hand, memory-based algorithms use the entire rating matrix to make recommendations. In order to do so, they use an aggregation measure by considering the ratings of the other users [17] (those most similar) for the same item. Different models can be obtained by considering different similarity measures and different aggregation criteria. Also *item-based* approaches, which take into account the similarity between items (two items are similar if they have been rated similarly) [18,19], appear as good alternatives to the user-based method.

On the other hand, in model-based algorithms, 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. This kind of model ranges from the classical Naive–Bayes [16,20,21] to the use of more sophisticated techniques such as those based on aspect models [22–25]. Aspect-based models have been proposed as an approach to recommendation, for robust handling of the item cold-start problem. These models do not attempt to directly model pairwise interactions, instead they assume a latent or hidden variable that represents the different topics.

A good survey of the application of different machine learning approaches to the problem of collaborative filtering is [4].

- *Hybrid recommender systems* combine collaborative and content information. Depending on the hybridization approach different types of systems can be found [9]. Firstly we are going to consider those approaches that require building separate recommender systems using techniques that are specialized to each kind of information used, and then combine the outputs of these systems. For instance, the resulting scores can be combined using a weighted approach [26] or voting mechanisms [27], switching between different recommenders [28,29], and filtering or reranking the results of one recommender with another [9]. A different approach consists of combining both content and collaborative features. By means of this combination a single unified technique might be used regardless of the types of information used [16,7]. In such systems, a careful selection of the features is needed.

There have been some works on using boosting algorithms for hybrid recommendations [30,31]. These works attempt to generate new synthetic ratings in order to alleviate the cold-start problem. These new ratings can be obtained using various heuristics, based on content information (for instance according to who acted in a movie) or demographic information. After injecting these new ratings into the user-item matrix along with actual user ratings, a collaborative algorithm is used.

The use of aspect models [24] has been also extended to use many types of meta-data (e.g. actors, genres, and directors for movies) [32]. A similar approach has been also used for music recommendations [33] and online document browsing [34]. Also related, the hybrid Poisson-aspect model [35] approach combines a user-item aspect model with a content-based user cluster.

2.1. Canonical weighted sum: a gateway to solve complexity problems

Our hybrid proposal can be viewed as an extension of the BN-based collaborative model in [36], which will be discussed in more detail in this section. In terms of dependence relationships, this model considers that the active user's ratings are dependent on the ratings given by similar users in the system. The topology of the BN consists of a variable A, representing the active user, having as its parents those user variables, $U_i \in \mathcal{U}$, with most similar tastes. These parents are learned from the database of votes. As a similarity measure between the active user A and any other user U ($sim(A,U)$) a combination of two different but complementary criteria has been used: on the one hand, we use rating correlation (Pearson correlation coefficient, PC) between common items to measure the similarity between the ratings patterns. The second criterion attempts to penalize those highly correlated neighbours which are based on very few co-rated items, which have proved to be bad predictors [17]. In some way, we are measuring how the sets of ratings overlap:

$$sim(A,U) = abs(PC(A,U)) \times \frac{|I(A) \cap I(U)|}{|I(A)|}, \tag{1}$$

where I is the set of items rated by the user in the data set.[1] In our approach, by using the absolute value of PC, $abs(PC)$, we consider that both positively (those with similar ratings) and negatively correlated users (those with opposite tastes) might help to predict the final rating for an active user.

Taking into account the number of users involved in the predictions, in [36] we developed a canonical model to represent the conditional probability distributions: the *canonical weighted sum (CWS)* gate. When this model is assumed, we can factorize the conditional probability tables into smaller pieces (the weights describing the mechanisms) and use an additive criterion to combine these values. This model can be seen as an example of "independence of causal influence" [37,38], where causes lead to effect through independent mechanisms. Since the system presented in this paper also has to handle a large number of variables (users, items and features) we are going to briefly review this model.

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

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

where $y_{k,l}$ is the value that variable Y_k takes in the configuration $pa(X_i)$, and $w(y_{k,l},x_{i,j})$ are weights (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 for each configuration $pa(X_i)$

[1] It should be noted that we are not considering the particular votes, merely whether the users rated an item or not.

Table 1
Example of matrices containing product descriptions, $\mathcal{D}$, and user ratings, **S**.

$\mathcal{I}/\mathcal{F}$	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
I_1	1	1	1	0	0	0	0	0
I_2	1	1	1	0	0	0	0	0
I_3	0	0	1	1	1	0	0	0
I_4	0	0	0	1	1	1	0	0
I_5	0	0	0	1	1	1	0	0
I_6	0	0	0	0	0	1	1	1
I_7	0	0	0	0	0	1	1	1
I_8	1	0	1	0	0	0	1	1
I_9	0	1	0	0	1	0	0	1
I_{10}	0	0	0	0	1	1	0	1

$\mathcal{U}/\mathcal{I}$	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
U_1	5	5	3	1	3	3	0	0	0	0
U_2	5	5	4	1	1	4	0	0	0	0
U_3	4	4	3	2	2	4	0	0	0	0
U_4	1	0	1	0	0	2	1	3	0	5
U_5	1	0	2	0	0	1	1	3	0	3
U_6	0	2	1	0	2	2	0	2	0	0
U_7	0	0	5	5	0	2	0	0	0	4
U_8	5	4	3	3	0	2	1	2	1	0
U_9	0	0	1	1	2	2	0	0	0	2
U_{10}	0	0	5	0	4	0	0	0	0	3

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

For example, assume that X_i has four parents, $Pa(X_i) = \{Y_1, \ldots, Y_4\}$, and that each variable takes its values in $\{a,b,c\}$. Then, given the configuration $pa(X_i) = (y_{1,c}, y_{2,a}, y_{3,c}, y_{4,b})$, we shall compute $Pr(x_{i,b}|y_{1,c}, y_{2,a}, y_{3,c}, y_{4,b})$ as the sum $w(y_{1,c}, x_{i,b}) + w(y_{2,a}, x_{i,b}) + w(y_{3,c}, x_{i,b}) + w(y_{4,b}, x_{i,b})$.

By means of this model we can tackle efficiently those complexity problems related to probability estimation, storage and inference. Thus, it is possible to estimate large conditional probability distributions since it is only necessary to estimate the weights involved in computing the conditional probability distributions in Eq. (2). Various additional advantages will also be obtained: firstly, since these weights can be computed independently (taking only a pair of variables into account), we reduce the problem of data sparsity; secondly, the parent set of X_i can be easily modified (for instance, including a new variable Y_{k+1} as the parent of X_i does not involve recomputing every conditional probability value); and thirdly, the use of this canonical model allows us to design a very efficient inference procedure (see Section 4).

The CWS gate has its own limitations since a general probability distribution cannot be represented by means of this gate. It only can represent properly those situation where the joint distribution can be computed by adding the individual's weights. Nevertheless, we believe that its use is appropriate in the recommender framework.

3. General description of the hybrid recommender model based on Bayesian networks

In this section we will describe the BN used to represent the hybrid system. This model represents how users, $\mathcal{U}$, items, $\mathcal{I}$, and features, $\mathcal{F}$, are related. Focusing on the input data, the content description of the items is usually expressed by means of a sparse binary matrix, $\mathcal{D}$, of size $m \times l$, where $d_{i,j} = 1$ when item i is described by feature j. When the entry is null, this relation is not established. An example of such a matrix is presented on the left-hand side of Table 1. Similarly, the ratings are also represented by means of a matrix, $\mathcal{S}$, of size $n \times m$, where users are represented in the rows and items in the columns. This matrix is usually sparse as users usually rate a low number of items. The value of the matrix, $s_{a,j}$ represents how user U_a has rated item I_j. We denote by $\mathcal{R}$ the rating's domain. When a user has not rated a product, the value is 0. The right-hand side of Table 1 shows an example of such a matrix.

3.1. Elements in a recommender context

Since our BN-based system should include information about users $\mathcal{U}$, items $\mathcal{I}$ and features $\mathcal{F}$, we are going to consider the domain of these variables:

- Features nodes: There will be an attribute node F_k for each feature used to describe a product. Each node has an associated binary random variable which takes its values from the set $\{f_{k,0}, f_{k,1}\}$, which means that the kth feature is not relevant (not apply), $f_{k,0}$, or is relevant (apply), $f_{k,1}$, for the description of the content of a product. [2]

[2] In our framework the term "relevant" expresses that it can help to (it is relevant for) predicting the target item's ratings.

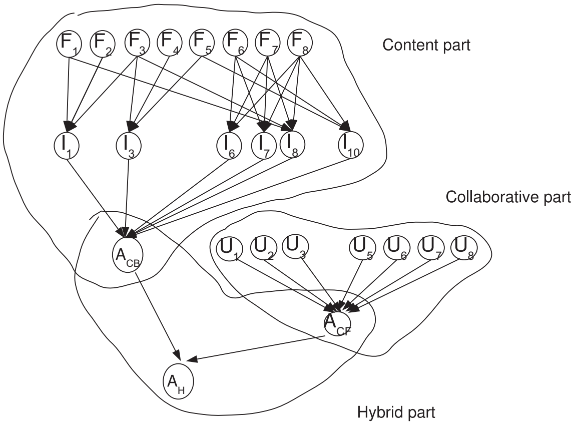


Fig. 1. The static subgraph of the hybrid Bayesian network.

- Item nodes: Similarly, there is a node I_j, for each item. The random variable associated with I_j will take its values from the set $\{i_{j,0}, i_{j,1}\}$ meaning that the item is not relevant (not apply) or is relevant (apply), respectively, when it comes to predicting the user's rating.
- User nodes: These U_i nodes will be used to predict the rating for the target item, particularly they should represent how probable is "the user rates with s an item". The domain of this variable is therefore the set $\mathcal{R} \cup \{0\}$. The additional value, 0, is included to model the lack of knowledge, i.e. the user has no useful information for predicting the target item's rating.

3.2. Topology of the model

Concerning the topology of our proposal, we shall distinguish two different parts: the first one is used to represent the knowledge that we have about how the active user would rate an item, i.e. the user profile. Since this component is centred on the user's perspective it might be static and could be built in an offline process. On the other hand, the second component represents those relationships related to the target item. As a consequence this part, which changes from one recommendation to another, has to be built dynamically. We are going to discuss these components in detail.

3.2.1. Static topology: representing the user profile

The user profile will be used to predict how the active user A would rate an item. In our hybrid approach, we use a BN to represent both content and collaborative components. Then, these components will be integrated in order to complete our hybrid system (see Fig. 1). The topology of the content part will be fixed (we only have to estimate the probability values from the data sets). We use a user variable A_{CB} gathering the information needed to perform content-based predictions. With respect to the collaborative component, we have to look for users similar to the active user (and therefore, a learning process becomes necessary). In this case, the collaborative information is also gathered in a user variable A_{CF}. To allow the combination of both components we use a variable A_H to encode the active user's predictions at the hybrid level. Following Burke's [9] ideas, the way in which we model the user profile can be considered as "mixed" since this variable encodes the mechanism controlling the contribution of both content and collaborative approaches.

Now, we are going to describe these components (for illustrative purposes, Fig. 1 shows the user profile associated to the user U_4, according with the data in Table 1):

CB *Content-based component:* We will consider that an item's relevance will depend on the relevance values of the features that define it. Therefore, there will be an arc from each feature node, F_i, to the nodes representing those items, I_j, which have been described with this feature. By directing the links in this way, we allow two items with a common subset of features to be dependent (except when we know the relevance values of these common features). For instance, using the data in Table 1, features F_1, F_3, F_7 and F_8 are connected to I_8.

In order to conclude this part, we must connect the nodes representing the items with the node representing the active user's predictions. The basic rule for performing these connections is simple: for each item I_j rated by the active user, add the arc $I_j \rightarrow A_{CB}$ to the graph.[3] Fig. 1 shows these arcs when the user U_4 plays the active user role.

CF *Collaborative component:* The collaborative component will comprise those people with similar tastes or preferences to the active user, represented by A_{CF}. These relations between users will depend on user ratings and so they must be

[3] The use of this model will imply that when the active user rates a new item, we must re-learn his or her conditional probability table. Nevertheless, by using the canonical weighted sum gate this process will be greatly simplified.

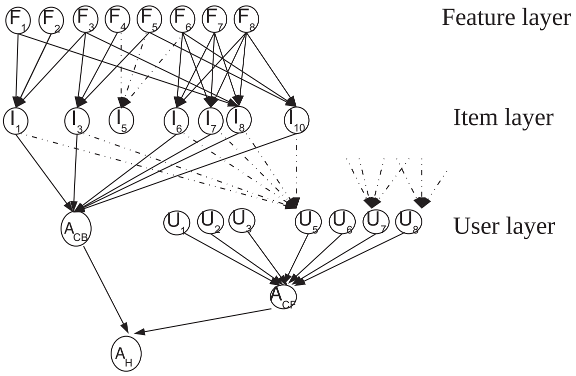


Fig. 2. Extending the static component with the item-dependent relationships.

learnt from the database of votes, **S**. In this paper we use as a similarity measure the criterion previously discussed in Section 2.1 (see Eq. (1)). Regardless of the mechanism used to find these relationships, whenever a dependence (similarity) between the preferences of the active user and a given user U_i is found, an arc connecting both nodes should be included in the Bayesian network, $U_i \rightarrow A_{CF}$.

Following a common approach in traditional memory-based RS, we use a fixed neighbourhood (selecting the top-N most similar users). This approach simply selects more users so that the predictions might be based on a sufficient number of ratings. As a consequence, not all users in the selected neighbourhood have given a rating for the item that we want to predict. An advantage of this approach is that we have a neighbourhood (profile) which is independent of the target item. Therefore, an offline learning algorithm can be applied in order to update the system after new users or ratings arrive. This update can be done when the system has a low workload.

H *Hybrid component:* Given an active user A, we will have his or her own preferences about the relevance of a new item in node A_{CB} (representing the content-based component), and also the preferences borrowed from similar users in node A_{CF} (collaborative component). These two preferences must be combined in order to obtain the final prediction for the user. This can be easily represented in the BN-based model by including a new node, A_H, which has both content (A_{CB}) and collaborative (A_{CF}) information as its parents.

3.2.2. Dynamic topology: managing target item-dependent relationships

Given the active user profile, the purpose of the model is to predict the rating of an unobserved item. In order to take into account the information associated with this target item, we can enlarge both, content and collaborative components. The content-based component is enlarged by inserting a new node which represents the item itself. This node will be linked with all the features used to describe the item. For example, Fig. 2 illustrates this situation when we are trying to predict how U_4 should rate item I_5. In this figure we use dashed lines to denote the dynamic relationships.

Focusing on the collaborative component, it is possible to distinguish between those users who rated the target item I in the past ($\mathcal{U}_I^+$) and those who did not ($\mathcal{U}_I^-$). In the first case, we know exactly what the given ratings were. In the latter case, a first alternative might be not to use any information related to the users in $\mathcal{U}_I^-$ when recommending. This is common in a pure collaborative context since we do not have any more information. In a hybrid context, however, we might think about the use of content-based information in order to get some knowledge about how these users in $\mathcal{U}_I^-$ should rate an item. This kind of information can be included easily in our model by connecting each user U_i in $\mathcal{U}_I^-$ with the set of items previously rated by U_i. Continuing with our example $\mathcal{U}_I^- = \{U_5, U_7, U_8\}$ because they did not rate I_5. For clarity, in Fig. 2 we only show the links representing this content-based information for the user U_5.

With this approach, we allow not only for the active user to receive content information when recommending but also that those similar users in $\mathcal{U}_I^-$ might be favoured with this type of information in the collaborative component. Using Burke's classification [10], our hybrid approach could therefore also be placed in the "Feature Augmentation" class since we are combining both content and collaborative features when computing the probabilities in A_{CF}.

In the next section we are going to describe how the inference can be performed. Then, in Section 5 we consider how the particular weights in the CWS are assessed. We expect that this ordering will help the reader to get a better understanding about the assessment of the conditional probability distributions.

4. Inference mechanism: computing recommendations

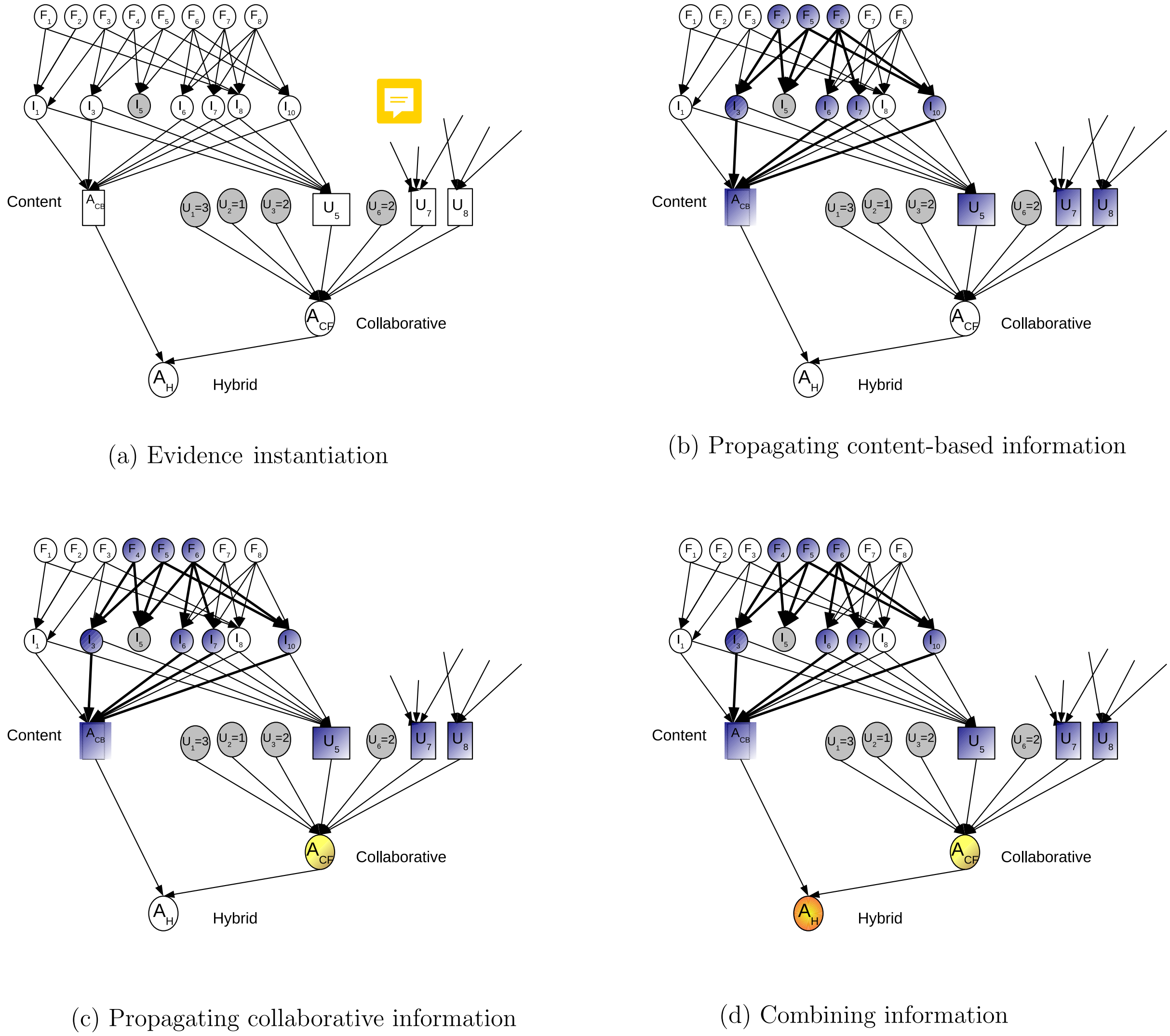
In this section we will see how the proposed topology and the use of canonical models to estimate the probability distributions enable very efficient inference mechanisms. Our goal is to compute how probable it is that the active user rates with a particular rating, given the evidence ev, i.e. $Pr(A_H = r|ev)$ for all $r \in \mathcal{R}$. In general terms, we have to instantiate the evidence in the BN and propagate towards the predictive nodes. Before studying how propagations should be performed, it is necessary to discuss how users should interact with the system.

4.1. Managing the evidence

In our framework, we will consider two different types of evidence given by content (ev_{cb}) and collaborative (ev_{cf}) information, i.e. $ev = ev_{cb} \cup ev_{cf}$.

- ev_{cb}: Focusing on the content component, we can consider two different approaches. On the one hand we can consider the item itself, in our example I_5, as evidence. So, we can instantiate this node I_5 to relevant, i.e. $ev_{cb} = \{i_{5,1}\}$. This approach will be denoted as *item instantiation*. The second alternative is to consider that the evidence comprises the features used to describe the item: F_4, F_5 and F_6 in the example. In this situation, we will instantiate all the features used to describe an item to relevant, $ev_{cb} = \{f_{4,1}, f_{5,1}, f_{6,1}\}$, and this is called *feature instantiation*.
- ev_{cf}: Focusing on the collaborative component, we know those users who rated the target item in the past, $\mathcal{U}_I^+$. Therefore, we can use the given rating as evidence. Continuing with our example, we know that I_5 was rated 3 by U_1, 1 by U_2, 2 by U_3, and 2 by U_6, i.e. the evidence is $ev_{cf} = \{u_{1,3}, u_{2,1}, u_{3,2}, u_{6,2}\}$.

Once the evidence is inserted in the model (Fig. 3a shows the instantiation of the evidence when predicting the rating for the item I_5), this information will be propagated through the network towards the predictive nodes.



(a) Evidence instantiation

(b) Propagating content-based information

(c) Propagating collaborative information

(d) Combining information

Fig. 3. Propagating the evidence towards predictive variables.

Table 2
Algorithm to compute $Pr(H_a|ev_{cb} \cup ev_{cf})$.

```
1. Content-based propagation:
   – If ($ev_{cb} == I_j$)// Item instantiation (see Fig. 3a)
        set $Pr(i_{j,1}|ev) = 1$
        Compute $Pr(F_k|ev)$ using Theorem 2//propagating towards features,
   else for each $F_k \in I_j$ set $Pr(F_k = 1|ev) = 1$.// Features Inst.
   – Propagate to items using Theorem 1.
   – Propagate to $A_{CB}$ and $U_i \in \mathcal{U}_I^-$ using Theorem 1.// (see Fig. 3b).
2. Collaborative propagation:
   – For each $U_k \in \mathcal{U}_I^+$ set $Pr(U_k = r_{k,j}|ev_{cf}) = 1$.// Collaborative evidence.
   – Propagate to $A_{CF}$ node using Theorem 1.// (see Fig. 3c)
3. Combine content-based and collaborative likelihoods at hybrid node $A_H$
4. Select the predicted rating.
```

4.2. Propagation processes

The aim of the inference process (as mentioned previously) is to estimate the rating of the active user A, given the evidence $Pr(A = s|ev)$. This propagation implies a marginalization process (summing out over uninstantiated variables) which requires an exponential time. Nevertheless, taking into account that:

(1) in a Bayesian network, a node is independent of all its ancestors given that the values taken by its parents are known,
(2) the conditional probabilities are represented using the canonical weighted sum gate (Eq. (2)),

the a posteriori probability distributions can be efficiently computed as a top-down inference mechanism. In abstract terms, the topology supporting the hybrid recommender model consists of three node layers (feature, item and user's layers) plus two more used to encode the active user predictions. Thus, starting from the first existing layer in the Bayesian network, the distributions of one layer are obtained using the a posteriori probabilities computed in the previous one. Fig. 3 illustrates the propagation process.

The following theorem (see [36]) explains how to compute the exact probability values. By means of this theorem, we express that each node collects the evidence from its predecessors and does not need to be distributed again. This is important because exact propagation can be performed in linear time with the number of parents (proof of this theorem can be found in the Appendix in [36]).

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. (2) *and the evidence is only on the ancestors of X_a, then the exact posterior probabilities can be computed using the following formula:*

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

Focusing on the content-based component, the evidence would either comprise a set of features for an item (evidence in the first layer of the Bayesian network) or the item itself (in the second layer). In the first case, propagation is carried out directly as explained in Theorem 1. In the second case (instantiating items), the probability $Pr(F_k|i_{j,1})$ must be computed for each feature node F_k linked to the target item I_j. These posterior probabilities can then be incorporated into the propagation process. The following theorem (the proof is straight-forward) shows how to compute these values.

Theorem 2. *Let F_k be a parent node of I_j in a Bayesian network, with the former being a root node in the network. The a posteriori probability of relevance of the feature given the variable I_j playing the role of evidence is then computed as follows:*

$$Pr(f_{k,1}|i_{j,1}) = \begin{cases} Pr(f_{k,1}) & \text{if } F_k \notin Pa(I_j) \\ Pr(f_{k,1}) + \frac{w(f_{k,1},i_{j,1})Pr(f_{k,1})(1-Pr(f_{k,1}))}{Pr(i_{j,1})} & \text{if } F_k \in Pa(I_j). \end{cases} \tag{3}$$

where $Pr(i_{j,1}) = \sum_{F_K \in Pa(I_j)} w(f_{k,1}, i_{j,1}) Pr(f_{k,1})$.

The algorithm in Table 2 explains how the propagation process can be performed. In this algorithm, we consider that if U_k is a user who previously rated the target item I_j, then $r_{k,j}$ is the given rating.

5. Estimation of probability distributions

In order to complete the model's specification, the numerical values for the conditional probabilities must be estimated from the data sets. One important point to be considered is related to the size of the distributions that must be stored in a Bayesian network, exponential with the number of parents. Therefore, the assessment, storage and use of these large probability distributions can be quite complex.

As we said, we will use the canonical weighted sum model (see Section 2.1) to model item and user variables. When this model is assumed, we factorize the conditional probability tables into a set of weights and use an additive criterion to combine these values. We will now present various methods for estimating these weights.

(1) $\mathcal{F}$ *Feature variables:* Starting from the feature nodes (as these do not have parents) it is only necessary to compute the a priori probability distributions of relevance. In this paper, we propose two different alternatives for estimating these probabilities:
- *EP*: all features being equally probable, i.e. $Pr(f_{k,1}) = \frac{1}{l}$.
- *RF*: relative frequency, i.e. $Pr(f_{k,1}) = \frac{n_k+0.5}{m+1}$.

where l is the size of the set $\mathcal{F}$, n_k the number of times that feature F_k has been used to describe an item, i.e. the column sum of the left-hand side of Table 1, and m the number of items. The value $Pr(f_{k,0})$ is obtained as $Pr(f_{k,0}) = 1 - Pr(f_{k,1})$.

(2) $\mathcal{I}$ *Item Variables:* With respect to the item nodes, $I_j \in \mathcal{I}$, as these represent a binary variable, the only weights to be defined are those needed to compute $Pr(i_{j,1}|pa(I_j))$, since $Pr(i_{j,0}|pa(I_j)) = 1 - Pr(i_{j,1}|pa(I_j))$.

In order to assess these values we will consider the following idea: Assume that F_1 and F_2 are two features describing an item I_j, with F_1 being a common feature (in the sense that it has been used to describe many items) and F_2 a rare feature (it appears in few items). It is natural to think that when both features are relevant ($F_1 = f_{1,1}$ and $F_2 = f_{2,1}$) the contribution of F_2 on the I_j's relevance degree will be greater than the contribution of F_1. This idea has been widely used in the field of information retrieval [39,40] to consider the importance of a term in the entire document collection. Particularly, the concept of *inverted document frequency (idf)* is used to measure the term's importance. Therefore, using an *idf*-based approach we use the expression $\log((m/n_k) + 1)$ to measure the importance of a feature in the entire database. Obviously, when a feature is not relevant its weight is set to zero. Therefore, the weights will be computed as

$$w(f_{k,1}, i_{j,1}) = \frac{1}{M(I_j)} \log\left(\left(\frac{m}{n_k}\right) + 1\right) \quad \text{and} \quad w(f_{k,0}, i_{j,1}) = 0 \tag{4}$$

with $M(I_j)$ being a normalizing factor computed as

$$M(I_j) = \sum_{F_k \in Pa(I_j)} \log\left(\left(\frac{m}{n_k}\right) + 1\right). \tag{5}$$

For example, considering the item I_6 in Table 1 we have that $w(f_{6,1},i_{6,1}) = 0.3$, $w(f_{7,1},i_{6,1}) = 0.4$ and $w(f_{8,1},i_{6,1}) = 0.3$. Thus, using the CWS (Definition 1) we have that $Pr(i_{6,1}|f_{6,1},f_{7,1},f_{8,1}) = 1$, $Pr(i_{6,1}|f_{6,0},f_{7,1},f_{8,1}) = 0.7$,$Pr(i_{6,1}|f_{6,0},f_{7,1},f_{8,0}) = 0.4$ and so on.

(3) $\mathcal{U}$ *User variables*: In this case we have to distinguish between those variables representing the content-based predictions (having items as their parents, i.e. A_{CB} and $U_i \in \mathcal{U}_I^-$) and the variable that combines collaborative information (having users as their parents, i.e. A_{CF}). Note that for those users in $\mathcal{U}_I^+$ the given rating is known, and therefore no probabilities have to be estimated.
- *Content-based predictions*: In this case, we must consider the influence of an item in the rating pattern of the user. To assess these weights we will consider two criteria: Firstly, for a given user $U_{CB} \in \{A_{CB}\} \cup \mathcal{U}_I^-$, whenever he or she rated an item I_k with the value s, then all the probability mass should be assigned to the same rating s at the user level. For example, since U_4 rates I_6 with 2 (see Table 1) we have that when I_6 is relevant, i.e. $I_6 = i_{6,1}$, then all the probability mass must be sent to the state (rating) 2 at the user node, $U_4 = u_{4,2}$. On the other hand, we will assume that all the items are equally important for predicting the active user's rating. Thus, taking into account these two ideas, and depending on whether the item I_k appears as relevant or not in the configuration $pa(U_{CB})$ ($I_k = i_{k,1}$ or $I_k = i_{k,0}$, respectively), these weights might be defined as follows:

$$\begin{aligned}
&w(i_{k,1}, u_{cb,s}) = \frac{1}{|I(U_{CB})|},\\
&w(i_{k,1}, u_{cb,t}) = 0, \quad \text{if } t \neq s, \quad 0 \leqslant t \leqslant \#r,\\
&w(i_{k,0}, u_{cb,0}) = \frac{1}{|I(U_{CB})|},\\
&w(i_{k,0}, u_{cb,t}) = 0, \quad \text{if } 1 \leqslant t \leqslant \#r.
\end{aligned} \tag{6}$$

Note that when an item is not relevant for predicting purposes, $I_k = i_{k,0}$, all the probability mass is assigned to the state 0 at the user level, representing the lack of knowledge. Thus, continuing with the example, $w(i_{6,1}, u_{4,2}) = 0.166$, $w(i_{7,1}, u_{4,1}) = 0.166$ and so on. Then, for example, given the configuration $pa(U_4) = \{i_{1,1}, i_{3,1}, i_{6,1}, i_{7,1}, i_{8,1}, i_{10,1}\}$ we have that $Pr(u_{4,1}|pa(U_4)) = 0.166 + 0.166 + 0.166 = 0.5$, $Pr(u_{4,2}|pa(U_4)) = 0.166$, $Pr(u_{4,3}|pa(U_4)) = 0.166$ and $Pr(u_{4,5}|pa(U_4)) = 0.166$. Note that when all the items are considered to be "relevant" (as before) the estimated distribution corresponds with the one that might be obtained using the maximum likelihood estimator from the U_4's ratings. This is because the assumption of "independence of the causal influences" holds. Similarly if $pa(U_4) = \{i_{1,0}, i_{3,0}, i_{6,1}, i_{7,1}, i_{8,0}, i_{10,0}\}$, we will have that $Pr(u_{4,2}|pa(U_4)) = 0.166$, $Pr(u_{4,1}|pa(U_4)) = 0.166$ and that $Pr(u_{4,0}|pa(U_4)) = 0.666$.

- A_{CF} *Collaborative-based predictions*: In this case, we must determine the weights reflecting the contribution of each similar user U_i in the prediction of the rating for the active user A. We will use similar ideas as in [36], but taking into account also the probability mass associated with the lack of knowledge at the parent nodes, i.e. the probabilities associated to the state $u_{i,0}$. To a certain extent, this mass captures the uncertainty about the rating to recommend. The particular weights are:

$$\begin{aligned}
&w(u_{i,t}, a_{cf,s}) = RSim(U_i, A) \times Pr^*(A = s|U_i = t) \quad \text{if } 1 \leqslant t, s \leqslant \#r,\\
&w(u_{i,t}, a_{cf,0}) = 0 \quad \text{if } 1 \leqslant t \leqslant \#r,\\
&w(u_{i,0}, a_{cf,0}) = RSim(U_i, A),\\
&w(u_{i,0}, a_{cf,s}) = 0 \quad \text{if } 1 \leqslant s \leqslant \#r,
\end{aligned} \tag{7}$$

As we can see, these weights have two components: on the one hand, we consider the relative quality (importance or similarity) of each parent in relation to the active user, defined as $RSim(U_i, A) = Sim(U_i, A)/\sum_{j \in Pa(A_{CF})} Sim(U_j, A)$; and on the other, we will consider the probability of A rating with a value s when U_i rated with t, $Pr^*(A = s|U_i = t)$. These probabilities are obtained from the data set of user ratings. In order to estimate these values we only consider those items which have been rated by both U_i and the active user A, i.e. the set $I(U_i) \cap I(A)$. Thus, $Pr^*(A = s|U_i = t) = \frac{N(u_{i,t}, a_s) + 1/\#r}{N(u_{i,t}) + 1}$ where $N(u_{i,t}, a_s)$ is the number of times from $I(U_i) \cap I(A)$ which have been rated t by U_i and also s by the active user A. In addition, $N(u_{i,t})$ is the number of items in $I(U_i) \cap I(A)$ rated with t by U_i.

In order to illustrate how these weights work, consider that the active user has three parents, U_x, U_y and U_z with $RSim$ values equal to 0.6, 0.3 and 0.1, respectively. Also assume that $Pr^*(A = 4|U_x = 5) = 0.9$, $Pr^*(A = 4|U_y = 1) = 0.5$ and $Pr^*(A = 4|U_z = 4) = 0.9$. Then, we have that $Pr(a_{cf,4}|u_{x,5}, u_{y,1}, u_{z,4}) = 0.78$ or that $Pr(a_{cf,4}|u_{x,0}, u_{y,1}, u_{z,4}) = 0.24$ and $Pr(a_{cf,0}|u_{x,0}, u_{y,1}, u_{z,4}) = 0.6$.

(4) A_H *Hybrid variable*: As A_H node has two parents, A_{CB} and A_{CF}, representing the content-based and collaborative predictions, we must assess the conditional probability values $Pr(A_H|A_{CB}, A_{CF})$. These probabilities represent how to combine both types of information when predicting the active user's rating for the item I_j.

It is well known that the performance of the collaborative system improves as the information used for making recommendations increases. Inversely, the prediction for new or rare items (rated by a low number of similar users) becomes more difficult [26]. Taking this fact into account we propose that a parameter α_j, $0 \leqslant \alpha_j \leqslant 1$, be used to control the contributions of each component. Note that this parameter can vary from one recommendation to another.

$$\begin{aligned}
&Pr(a_{h,s}|a_{cb,s}, a_{cf,s}) = 1\\
&Pr(a_{h,s}|a_{cb,s}, a_{cf,q}) = \alpha_j, \quad \text{if } q \neq s\\
&Pr(a_{h,s}|a_{cb,t}, a_{cf,s}) = 1 - \alpha_j, \quad \text{if } t \neq s\\
&Pr(a_{h,s}|a_{cb,t}, a_{cf,q}) = 0 \quad \text{if } t, q \neq s
\end{aligned} \tag{8}$$

Considering this equation, the higher the value of α_j, the greater the weights of the content-based nodes. For example, assigning $\alpha_j = 0$, the hybrid model tends to behave as a collaborative model, since only the information at the collaborative node A_{CF} is taken into account. With $\alpha_j = 1$, on the other hand, it will behave as a content-based model. With intermediate values, the recommendation may be performed by taking into account content-based and collaborative information, which gives expression to the hybrid model.

5.1. Determining the α parameter

In this section we will discuss the particular way in which this parameter is assessed. In the literature, a range of mechanisms to hybridize have been considered, from using a fixed value to a more sophisticated method which depends on the number of items rated by the active user [26,30,29]. Our initial hypothesis is that the parameter α_j might depend on our confidence on the results obtained in the collaborative component (which in some way depends on the number of parents of the active user A who have rated the item I_j in the past).

In order to illustrate our point of view, we will analyze the hybrid model in greater detail. Let U_i be a similar user who did not rate the target item, $U_i \in \mathcal{U}_I^-$. In this case, we use content-based information in order to predict how this user should rate I_j. This information is represented by $Pr(U_i = s|ev) = Pr(U_i = s|ev_{cb})$, with $s \in \mathcal{R} \cup \{0\}$. The state 0 represents the situation where we do not have information for recommending. For instance, if none of the items rated by U_i were relevant to the target item, we will have that $Pr(U_i = 0|ev_{cb}) = 1$. Moreover, looking at Eq. (7), this probability mass will be propagated towards the state 0 at the collaborative node, A_{CF}. In other words, the probability $Pr(A_{CF} = 0|ev)$ will reflect in some way how uncertain we are about the prediction at the collaborative level. For example, if all the similar users (parents) rated the item, we will have that $Pr(A_{CF} = 0|ev) = 0$ whereas this probability takes its maximum value when none of the similar users rated this item in the past.

Therefore, $Pr(A_{CF} = 0|ev)$ reflects our confidence degree in the collaborative recommendation, it can therefore be used to determine the extent to which we might consider each model when merging the recommendations. Taking into account that the higher the value of α_j, the greater the weights of the content-based nodes, in this paper we propose [4] to use $\alpha_j = Pr(A_{CF} = 0|ev)^2$.

6. Evaluation of the hybrid recommender model

This section establishes the evaluation settings (data set, evaluation measures and experimentation aims) and also presents the experimental results for the performance of the hybrid model.

6.1. Data sets

In terms of the test data set, we have decided to use MovieLens. It was collected by the GroupLens Research Project at the University of Minnesota during the seven-month period between 19th September 1997 and 22nd April 1998 with votes ranging from 1 to 5 stars (1 = `Awful`, 2 = `Fairly bad`, 3 = `It's OK`, 4 = `Will enjoy`, 5 = `Must see`) in a cinematographic context.

The data set contains 1682 movies rated by 943 users, and contains a total of 100,000 transactions on a scale of 1–5. In order to perform 5-fold cross validation, we have used the data sets u1.base and u1.test through u5.base and u5.test provided by MovieLens which split the collection into 80% for training and 20% for testing, respectively.

We decided to use MovieLens mainly for the following reasons: it is publicly available and has been used in many hybrid recommender systems. For these reasons, we believe that it is a good benchmark for our purposes. Moreover, it is especially interesting because it offers a content component, as the 1682 movies are classified into 18 genres (action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war and western). This allows us to perform recommendations by considering the content part of the data set.

In view of the fact that the movies included in the MovieLens database are only described using 18 genres, we have also extended the content description of the movies by means of the additional information provided by the *Internet Movie Database–IMDB–*.[5] More specifically, new information (e.g. directors, producers, plot keywords and cast) has been considered as part of the movie description in order to enrich it. With this expansion, the number of features increases from the original 18 to 17024. There are various movies from MovieLens that are not included in IMDB and these are only characterized by their genre. This extension is common practice when testing recommender systems with a significant amount of content. Some examples are [30,41,42] with the EachMovie dataset or [43,44,6,29] with MovieLens.

Therefore, in this experimentation we consider two different data sets: the first includes only the genre description using the original MovieLens dataset (denoted by ML), and the second one uses an extended version which considers more features from IMDB (denoted by ML + IMDB).

6.2. Evaluation measures

With respect to the second decision, in order to test the performance of our model, we shall measure its capability to predict a user's true ratings or preferences, i.e. system accuracy. Following [45], we propose to use the mean absolute error (MAE) which measures how close 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.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} abs(p_i - r_i)}{N} \tag{9}$$

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.

6.3. Selecting the predicted rating

There is a key issue in a system's performance that has to be considered before presenting the experimental results. This issue consists of how to select one rating (the predicted rating) from a probability distribution over candidate ratings. There are several methods for computing this prediction. For instance, we can consider three different alternatives: the most probable rating, the expected rating and the median rating. Following [4], we will use the median prediction with all the models since it minimizes the mean absolute error.[6]

If we focus on the predictive variables in our models, A_{CB}, A_{CF} and A_H, we find that these variables take their values in $\mathcal{R} \cup \{0\}$. However, we must select a rating, *rate*, in $\mathcal{R}$. Before selecting the final rating, therefore, we must distribute the probability mass associated with the state zero, i.e. $Pr(A_\bullet = 0|ev)$. It should be remembered that this probability gathers all the mass associated with the lack of information in the recommending process. In this paper, we propose that a proportional

[4] We have also used a different approach with α being a function of the number of similar users who rated the item. This alternative gives worse results.

[5] http://www.imdb.com/.

[6] If the goal is to minimize the squared error, we should use the expected rating, and if the goal is to minimize the error rate, we should use the most probable rating.

Table 3
MAE values for *hybrid* model with MovieLens.

	ML (I_I)		ML(I_F)		ML + IMDB(I_I)		CF
NS	EQ	RF	EQ	RF	*EQ*	RF	
10	0.7293	0.7360	0.7813	0.7878	0.7254	0.7285	*0.7579*
20	0.7307	0.7393	0.7807	0.7876	**0.7198**	0.7277	0.7637
30	0.7330	0.7422	0.7808	0.7872	0.7207	0.7292	0.7681
50	0.7364	0.7467	0.7802	0.7879	0.7231	0.7328	0.7735
75	0.7401	0.7505	0.7804	0.7875	0.7252	0.7355	0.7784
CB	0.7837	0.7892	*0.7833*	0.7857	0.7908	0.7975	

criterion be used.[7] For a given active user A, we transform the posterior probability into a new one in the domain $\mathcal{R}$ using the following expression

$$Pr(A_\bullet = s|ev) = \frac{Pr(A_\bullet = s|ev)}{1 - Pr(A_\bullet = 0|ev)}, \quad \forall s \in \mathcal{R}.$$

Once we have the posterior probabilities in $\mathcal{R}$, the predictions (*rate*) for the active user A is the median rating:

$$rate = \{s|Pr(A_\bullet \leqslant s|ev) \geqslant 0.5,\ Pr(A_\bullet > s|ev) \geqslant 0.5.\} \tag{10}$$

6.4. Results for the hybrid model

In this section, we present the results obtained by the hybrid recommender model. The aim of this experimentation is to determine the validity of our approach and also to study the contribution of each component in the recommendation. In order to achieve this aim we have considered the following experimental conditions (Table 3 shows the obtained results):

Firstly, we have considered both MovieLens (ML) and the extension using IMDB (ML + IMDB) data sets. Secondly, we distinguish between item instantiation, I_I, and feature instantiation, I_F (see Section 4.1). Thirdly, we have considered the two different a priori values in the features nodes. The first, where all the features are equally probable, *EQ*, and the second which considers how frequently a feature has been used to describe a movie, *RF* (see Section 5). Finally, we also report the results obtained using different neighbourhood sizes (NS). To test the sensibility of the model with respect to the neighbourhood sizes, we have considered the most 10, 20, 30, 50 and 75 similar users.

The last row in Table 3 presents the results obtained by considering only the content-based component, i.e. the results obtained by predicting the rating using the probabilities in A_{CB}. Similarly, the last column presents the results obtained by considering *pure* collaborative information, i.e. the results obtained by first considering for all $U_i \in \mathcal{U}_I^-$ that $Pr(U_i = 0|ev_{cb}) = 1$ and then predicting the rating using the probabilities in A_{CF}.

Focusing on content-based predictions, we should highlight the worsening of performance of ML + IMDB in relation to the experiments using the original MovieLens dataset. The reason for this is clear since the efficiency of a content-based recommender system (which is in fact an information retrieval system) worsens when the number of terms increases, mainly because of the inclusion of non-significant features, as in the case of cast (i.e. IMDB considers all the actors and actresses with a role in the movie).

Focusing on collaborative predictions we can also observe that using a small number of neighbours tends to result in greater prediction accuracy. We should mention that this situation also appears when using a classical neighbour-based approach [17]. It seems that if there are many parents, some noise is introduced and the performance of the model is damaged. Nevertheless, a precision/recall tradeoff exists when using a small number of parents because the number of ratings predicted without collaborative information increases.

Now we will discuss the results obtained by the hybrid model. From the data in Table 3 we can see that the hybrid model using item instantiation performs much better than feature instantiation.[8] Moreover, with this combination the hybrid model outperforms the results obtained using collaborative or content-based recommendations isolately. Comparing these results with those obtained with our baselines, we can appreciate that significant improvements of around 6–8% have been achieved. In relation to the probabilities stored at the feature nodes, we can observe that it is better to consider that all the features are equally probable a priori. From these results it can be concluded that it is not very important that the recommendation process considers how relevant is a given feature to describe the target item.

Also, in a similar way to the pure collaborative model, we obtain better results using a small number of parents. The performance of the model worsens, in general, when the size of the parent set increases. This is true when considering item instantiation, but the performance is stable in the case of feature instantiation.

[7] We have also considered assigning the entire mass to the rating with the greatest posterior probability, but worse results were obtained.

[8] This also holds when considering ML + IMDB data set; accuracy using feature instantiation was similar to that obtained using only the genres (ML) with MAE values of around 0.785.

To conclude, under the same experimental conditions, the best results were obtained when the content description is extended with IMDB (ML + IMBD) data set. Taking into account that using IMDB worsens the performance in the content-based approach, we guess that the use of extra features might improve the recommendations in the collaborative component (via the variables in $\mathcal{U}_I^-$) and that these improvements are better in those items where the collaborative filtering does not work well, i.e. those items where the cold-start is an issue. Since usually these are rare items, the use of extra information seems to be beneficial. In order to corroborate this guess, we have run some experiments using content information at the nodes in $\mathcal{U}_I^-$. For example using I_I, EQ, $NS = 20$, and predicting the rating at the collaborative variable A_{CF} we have obtained a mean MAE of 0.7362. Comparing this result with the one obtained using the pure collaborative model (0.7637) we have that a significant improvement is obtained by using content-based information. Finally, if we compare this result with the 0.7198 in Table 3 (obtained at A_H variable), we might conclude that this last improvement (from 0.7362 to 0.7198) is mainly due to the way in which both content and collaborative information are mixed at the hybrid node A_H.

To conclude, we talk about the efficiency of the model. In this case, we say that the propagation process 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), it is only necessary to compute those probability values affected by the evidence. Therefore, we only have to compute the posterior probabilities for those variables in the path from evidence nodes to the respective active user nodes. Moreover, since the computations of the necessary posterior probabilities at each node are linear with the number of parents, they can be computed efficiently. Thus, we conclude that the model is appropriate for being used in many real world applications when users are logged onto the system online, even when there is a large database of ratings.

6.5. Comparison with other models

In terms of the comparison of the performance of our proposal with other systems, it is worth mentioning that it is extremely difficult to find papers where the experimental setting is the same or at least reproducible in the context of hybrid systems. While there are changes in the goals of the papers, the sources of content data (differing on the use of product description, the use of social and/or demographic data about users) and also the way that training and testing, movie selection, user selection, feature selection, etc. are carried out. Therefore, and in order to compare the results of our model we have implemented several hybrid, collaborative or content-based recommender systems. Table 4 presents the MAE results obtained by each system. In this table, the row entitled with % presents the improvement percentage obtained with our model.

Firstly, we consider the hybrid model in [29] (noted by *switch* in Table 4) since its experimentation is similar to ours. In this model, a user-based collaborative filtering approach is used as the primary method and switches to content-based when collaborative predictions cannot be made (the number of neighbours for the collaborative model is fewer than five users).

We have performed a similar experiment, noted by *BN-switch* in Table 4, switching between our pure content-based and collaborative recommendations (A_{CB} and A_{CF} nodes) following the same criteria as [29], i.e. we selected the content-based recommendation when fewer than five users rated the movie and the collaborative recommendation otherwise. We have fixed the following experimental conditions: ML + IMDB, EQ, I_I and 20 parents.

We also show the results obtained using the imputation-boosted collaborative filtering [30] (named IBCF in Table 4). Firstly, we use the predictions obtained with a content-based recommender system to fill-in the sparse user-item rating matrix. Particularly, following the ideas in [30], we have implemented a bag-of-words (features) naive Bayesian classifier. Then we run a traditional Pearson correlation-based CF algorithm on this complete matrix to predict a novel rating.

Finally, we have also considered a model-based hybrid approach [33] which is an extension of the three-way aspect model [34], denoted by *ModelH* in Table 4. This model explains the generative mechanism for both content and collaborative data by introducing a latent variable, which conceptually corresponds to user types. Particularly, the probability distribution over users, items and features is decomposed into three conditional independent ones by introducing the latent variable. The interpretation is that a user type is selected according to the active user's preferences and the item features, then the prediction is obtained by conditioning to the user type. We have tuned this model and we report the best result, obtained when using 6 different states for the latent variable.

In order to quantify the improvement of the hybrid model, we should compare the results with those obtained using different content-based and collaborative models separately. In the first case, we borrow the results obtained by two classical content-based predictors [29] using ML + IMDB: the first is the *pure content-based (PCB)* predictor in which the cosine

Table 4
MAE values for other models.

	Hybrid				Collaborative			Content	
	Switch	BN-switch	IBCF	*ModelH*	Ubased	*Ibased*	*Triadic*	PCB	NB
MAE	0.7501	0.7498	0.7544	0.7405	0.7654	0.7604	0.7365	0.9253	1.2434
%	4.2	4.1	4.8	2.9	6.3	5.6	2.3	28.5	72.7

measure is used to calculate the similarity between two items; and the second one is an implementation of the content-based model using a *Naive Bayes (NB)* algorithm where the ratings are considered as the class labels.

With respect to collaborative filtering, we have used three different algorithms: firstly, we compare our baseline with the most classical *user-based* collaborative filtering model [17]. In this model, the similarity between users is computed using Pearson's correlation coefficient. In addition, the contribution of neighbours with fewer than 50 commonly-rated movies has been devaluated by applying a significance weight of $n/50$, where n is the number of ratings in common [17].

Also *item-based* approaches [18,19] appear as good alternatives to the user-based method. Item-based approaches look into the set of items the active user has rated and computes how similar they are to the target item, selecting the set of k most similar items. This item-based similarity is computed taking into account the ratings given by those users who have rated both of these items. Then the prediction is computed by taking a weighted average of the target user's ratings on these similar items. Particularly, in our experimentation we have used the adjusted cosine measure [18] and the result reported in Table 4 has been obtained when considering a neighbourhood size of 75.

Finally, we have considered the *Triadic* aspect model [23] (see Table 4) for pure collaborative predictions that considers a latent variable relating the triplet (user, item, rating). The purpose is to automatically look for the potential reasons that determine which subset of the causes are likely to be relevant for a specific item or a specific person, and in each individual case assigns a probability to the fact that a cause will be active for a given rating. If we compare this result with the ones obtained using *ModelH* we can see that extending the aspel model with content information does not lead to improvements. This result is similar to the one obtained in [32].

After evaluating these results, we could conclude that our model, reaching a best MAE of 0.7198, is competitive with the standards in the literature, producing better measures of quality.[9] We have also the advantage that these results are obtained using the same paradigm and that these facts might be exploited in order to give better explanations of the recommendations to the users.

7. Conclusions and further research

In this paper, we have proposed a hybrid recommender model based on Bayesian networks which uses probabilistic reasoning to compute the probability distribution over the expected rating. The model is founded on a layered topology representing all the elements involved in the hybrid recommendation problem. The participation degree of each recommending mechanism (content-based and collaborative) is automatically selected, adapting the model to the specific conditions of the problem. We have proved empirically that the combination of both content and collaborative information helps to improve the accuracy of the model.

Focusing on the computational aspects of the recommender process, problems such as data sparseness and the fact that the ranking should be computed in real time have been considered. In particular, guidelines for how to estimate the probability values from a data set are presented and an efficient propagation algorithm based on canonical models has also been designed.

Following Burke's classification [10], our hybrid approach could therefore be placed in the "Feature Augmentation" class since in normal operation the probabilities obtained in the propagation of the variable layers involved in a content-based recommendation are used to propagate probabilities in the layers relating to the collaborative recommendation. Moreover, as there is a mechanism to control the contribution of both elements, it may also be classified as "mixed".

It should be noted that the proposed model is versatile: it could work by exclusively applying content-based or collaborative filtering and it can also be applied to solve different recommendation tasks (such as *finding good items* or *predicting ratings*).

In terms of future research, we believe that there is room for improvement of the hybrid recommender model since there are several points that must still be researched:

- Design of new methods for estimating the weights stored in the nodes of the Bayesian network.
- Design of new feature selection methods (or the use of existing ones) to select only the best features so that the best performance may be achieved.
- Incorporation of relationships between features – this would involve introducing data mining techniques to find those features which might be related in terms of the classic co-occurrence measure of any other technique and would improve the expressiveness of the model.
- Change of the type of canonical model used in probability estimation and subsequently in the propagation-since the model uses sum gates, we could explore the possibility of applying either *And* or *Or* gates.

In the future, we therefore plan to study problems such as how our system can communicate its reasoning to users, the minimum amount of data (ratings or textual information) required to return accurate recommendations, and a more elaborate way of including item information.

[9] In order to show the variability in the conclusions we present the MAEs values per fold for the best model of our proposal (0.7304; 0.7206; 0.7069; 0.7201 and 0.7209) and the Triadic aspect model (0.75; 0.7369; 0.7306; 0.7328 and 0.7324).

References

[1] P. Resnick, H.R. Varian, Recommender systems, Communications of the ACM 40 (3) (1997) 56–58.
[2] S. Kangas, Collaborative filtering and recommendation systems, in: VTT Information Technology, 2002.
[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.
[4] B. Marlin, Collaborative Filtering: A Machine Learning Perspective, Master's thesis, University of Toronto, 2004.
[5] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37–51.
[6] Q. Li, B. Kim, Clustering approach for hybrid recommender system, in: IEEE/WIC Proceedings of the International Conference on Web Intelligence, 2003, pp. 33–38.
[7] A. Gunawardana, C. Meek, A unified approach to building hybrid recommender systems, in: RecSys'09: Proceedings of the third ACM Conference on Recommender Systems, 2009, pp. 117–124.
[8] P. Melville, R. Mooney, R. Nagarajan, Content-boosted collaborative filtering, in: ACM SIGIR 2001 Workshop on Recommender Systems, 2001.
[9] R. Burke, Hybrid recommender systems: survey and experiments, User Modeling and User-Adapted Interaction 12 (4) (2002) 331–370.
[10] R.D. Burke, Hybrid Web recommender systems, Lecture Notes in Computer Science 4321 (2007) 377–408.
[11] C. Butz, Exploiting contextual independencies in web search and user profiling, in: Proceedings of the World Congress on Computational Intelligence, 2002, pp. 1051–1056.
[12] R.J. Mooney, L. Roy, Content-based book recommending using learning for text categorization, in: DL'00: Proceedings of the Fifth ACM Conference on Digital Libraries, ACM Press, 2000, pp. 195–204.
[13] M. Pazzani, D. Billsus, Learning and revising user profiles: the identification of interesting web sites, Machine Learning 27 (1997) 313–331.
[14] 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.
[15] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, Generalizing e-bay.net: an approach to recommendation based on probabilistic computing, in: First Workshop on Web Personalization, Recommender Systems and Intelligent User Interface, 2005, pp. 24–33.
[16] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: 14th Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
[17] J. Herlocker, J. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: SIGIR G 99: Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 230–237.
[18] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the ACM World Wide Web Conference, 2001, pp. 285–295.
[19] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing 17 (6) (2003) 734–749.
[20] K. Miyahara, M.J. Pazzani, Collaborative filtering with the simple Bayesian classifier, in: Pacific Rim International Conference on Artificial Intelligence, 2000, pp. 679–689.
[21] V. Robles, P. Larrañaga, J. Peña, O. Marbán, J. Crespo, M. Pérez, Collaborative filtering using interval estimation Naive Bayes, in: Lecture Notes in Artificial Intelligence, vol. 2663, 2003, pp. 46–53.
[22] T. Hofmann, J. Puzicha, Latent class models for collaborative filtering, in: 16 Interantional Joint Conference on Artificial Intelligence, 1999, pp. 688–693.
[23] T. Hofmann, Learning what people (donG t) want, in: Machine Learning: ECML '01: Lecture Notes in Computer Science 2167, 2001, pp. 214–225.
[24] T. Hofmann, Latent semantic models for collaborative filtering, ACM Transactions on Information Systems 22 (1) (2004) 89–115.
[25] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: SIGIR '02: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp. 253–260.
[26] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin, Combining content-based and collaborative filters in an online newspaper, in: SIGIR99: Proceedings of the ACM SIGIR Workshop on Recommender Systems, 1999.
[27] M.J. Pazzani, A framework for collaborative, content-based and demographic filtering, Artificial Intelligence Review 13 (5–6) (1999) 393–408.
[28] D. Billsus, M.J. Pazzani, User modeling for adaptive news access, User Modeling and User-Adapted Interaction 10 (2–3) (2000) 147–180.
[29] G. Lekakos, P. Caravelas, A hybrid approach for movie recommendation, Multimedia Tools and Applications (36) (2008) 55–70.
[30] P. Melville, R. Mooney, R. Nagarajan, Content-boosted collaborative filtering for improved recommendations, in: Eighteenth National Conference on Artificial Intelligence (AAAI-02), 2002, pp. 187 – 192.
[31] S.-T. Park, D. Pennock, O. Madani, N. Good, D. DeCoste, Naïve filterbots for robust cold-start recommendations, in: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 699–705.
[32] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Generative models for cold-start recommendations, in: Proceedings of the 2001 SIGIR Workshop on Recommender Systems, 2001.
[33] K. Yoshii, M. Goto, K. Komatani, T. Ogata, H.G. Okuno, An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model, IEEE Transaction on Audio, Speech and Language Processing 16 (2) (2008) 435–447.
[34] A. Popescu, L. Ungar, D. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, in: 17th Conference on Uncertainty in Artificial Intelligence, 2001, pp. 437–444.
[35] C.-N. Hsu, H.-H. Chung, H.-S. Huang, Mining skewed and sparse transaction data for personalized shopping recommendation, Machine Learning 57 (1–2) (2004) 35–59.
[36] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, A collaborative recommender system based on probabilistic inference from fuzzy observations, Fuzzy Sets and Systems 159 (12) (2008) 1554–1576.
[37] C. Meek, D. Heckerman, Structure and parameter learning for causal independence and causal interaction models, in: Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 366 – 375.
[38] N.L. Zhang, D. Poole, Exploiting causal independence in Bayesian network inference, Journal of Artificial Intelligence Research 5 (1996) 301–328.
[39] C.J.V. Rijsbergen, Information Retrieval, second ed., Butterworth, London, UK, 1979.
[40] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1983.
[41] J. Basilico, T. Hofmann, Unifying collaborative and content-based filtering, in: ICML '04: Proceedings of the twenty-first International Conference on Machine Learning, 2004, pp. 9–16.
[42] K. Jung, D. Park, J. Lee, Hybrid collaborative filtering and content-based filtering for improved recommender system, Lecture Notes in Computer Science 3036 (2004) 295–302.
[43] N. Good, J.B. Schafer, J.A. Konstan, A. Borchers, B.M. Sarwar, J.L. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: Conference of the American Association for Artificial Intelligence, 1999, pp. 439–446.
[44] B. Bezerra, F. de Carvalho, A symbolic hybrid approach to face the new user problem in recommender systems, Lecture Notes in Artificial Intelligence 3339 (2004) 1011–1016.
[45] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (1) (2004) 5–53.