Soft Comput (2010) 14:785–798
DOI 10.1007/s00500-009-0474-5

FOCUS

Using second-hand information in collaborative recommender systems

L. M. de Campos · J. M. Fernández-Luna · J. F. Huete · M. A. Rueda-Morales

Published online: 30 July 2009

Abstract Building recommender systems (RSs) has attracted considerable attention in the recent years. The main problem with these systems lies in those items for which we have little information and which cause incorrect predictions. One accredited solution involves using the items' content information to improve these recommendations, but this cannot be applied in situations where the content information is unavailable. In this paper we present a novel idea to deal with this problem, using only the available users' ratings. The objective is to use all possible information in the dataset to improve recommendations made with little information. For this purpose we will use what we call second-hand information: in the recommendation process, when a similar user has not rated the target item, we will guess his/her preferences using the information available. This idea is independent from the RS used and, in order to test it, we will employ two different collaborative RS. The results obtained confirm the soundness of our proposal.

Keywords Collaborative recommender systems · Bayesian networks · Neighbourhood · Second-hand information

L. M. de Campos · J. M. Fernández-Luna · J. F. Huete · M. 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, 18071 Granada, Spain
e-mail: mrueda@decsai.ugr.es

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

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

J. F. Huete
e-mail: jhg@decsai.ugr.es

1 Introduction

The Internet has become an indispensable tool for our day-to-day lives. Not many years ago, when seeking information on any subject, we had to resort to the use of encyclopedias, printed magazines, public libraries, etc. Since the rise of the Internet, access to information is much faster and easier. However, what appeared to be an advantage has become a big problem, due to the vast amount of information that exists. In an attempt to solve this problem, automated tools have been popularized to help users find the items they seek. Recommender systems (RSs) have emerged to address this issue. In general, a RS provides specific suggestions regarding items (or actions) within a given domain and which may be considered of interest to the user (Resnick and Varian 1997). There are different types of RSs (Resnick and Varian 1997; Kangas 2002) depending on the information that is used to make the recommendation:

- Content-based RSs: recommend similar items to those the user has rated positively in the past.
- Collaborative RSs: identify groups of people with similar tastes to active user and recommend those items they liked.

In the present paper we will focus on collaborative options, as they are very efficient and are easy to implement and to adapt to real systems. Collaborative filtering techniques match people with similar preferences in order to make recommendations. Their aim is to predict the utility of items for a particular user according to the items previously evaluated by other users. The big advantage of collaborative approaches in comparison with content-based ones is their outside the box recommendation ability (Burke 2002), i.e., the possibility of recommending items that do not evince content features expressed in the user profiles. For example, it may occur that listeners who enjoy hard rock also enjoy flamenco music, but a content-based recommender trained on the preferences of a hard rock fan would not be able to suggest items in the flamenco realm, since none of the features (performers, instruments, repertories) associated with items in the different categories would match. Only by looking outside the preferences of the individual, such suggestions can be made.

The success of these systems depends on the availability of a critical mass of information. The problem arises when, given an active user, the system requires information on a specific item, and people with similar tastes are not capable of offering this. Thus, in these situations, the system will offer a prediction that simply is not good enough.

One possible approach found in the literature with regard to solving this problem involves the use of content-based approaches. In this case, when no collaborative information is available, predictions are computed using the items' content description (Popescul et al. 2001; de Campos et al. 2006; Degemmis et al. 2007; Ali and van Stam 2004). Another possible solution found is to make use of the content information to fill the missing ratings in the dataset and then use it in a collaborative recommendation (Melville et al. 2001; Su et al. 2008).

These approaches obviously depend on the availability of content descriptions. In this paper, we explore a new approach to tackle this problem which can be used in situations where content description is unavailable: what we have called second-hand information. In order to illustrate the idea, let us assume the following situation: John asks his friends for their opinion about a particular movie but none, or few of them have seen it. In an attempt to provide their own opinion of the movie, his friends decide to ask their friends. This is what we call second-hand information. In this paper, we will study whether these second-hand opinions can be used to improve the recommendation given to John. This idea can be implemented easily: For similar users who did not rate the target item in the past, we will use the rating that the system might predict for them, making use of the information from their own similar users.

In order to test our proposal, we used two RSs. The first one is a classical neighbour-based and the second one is Bayesian-based. Through different experiments, we have shown the beneficial effect that the incorporation of new quality information has on the behaviour of the systems. To compare the results, we evaluated an imputation-boosted (Su et al. 2008) model.

The rest of the paper is organized in the following manner: some background information on RSs is presented in Sect. 2. In Sect. 3, we will discuss how to obtain the second-hand information. In Sect. 4, we will explain the models we will work with and will subsequently present the results achieved in some experiments in Sect. 5. In Sect. 6, we will discuss the accomplishments of our approach and finally present some conclusions and possible future work in Sect. 7.

2 Recommender systems

Recommender systems help people to find products they are interested in and which they would otherwise not be aware of, due to the vast amount of information on the Internet. Depending on the information used to make the recommendation, there are different types of RSs (Resnick and Varian 1997; Kangas 2002):

- Content-based RSs: recommend similar items to those the user has rated positively in the past. The content-based RS are rooted in Information Retrieval (Belkin and Croft 1992) and use many of its techniques. Their underlying philosophy can be summed up by "recommend me items similar to those that I liked in the past". In content-based RS, items of interest are defined by their associated features, such as actors, directors, producers, genres, etc, in a movie recommendation system.
- Collaborative RSs: recommend the items that other users with similar tastes considered to be good. Broadly speaking, for each user, we obtain a set of users (his/her neighbours) with a rating pattern that is highly correlated with him/her. Thus, given an item not rated by the user, we can predict a rating for it based on a combination of the known values given to the item by his/her neighbours. In this paper we will focus upon this type of RS. Collaborative RS are also known as collaborative filtering systems.
- Hybrids: the recommendation is made by combining collaborative and content-based approaches.

As in this paper we will focus on collaborative RSs, we will extend the explanation of this type of system: According to Breese et al. (1998), collaborative RSs can be grouped into memory-based and model-based approaches. Memory-based algorithms use the entire rating matrix to make recommendations. In order to do so, they use some kind of aggregation measure by considering the ratings of other users (those most similar) for the same item. Different models can be obtained by considering different similarity measures and different aggregation criteria (Konstan et al. 1997; Herlocker et al. 1999).

In model-based algorithms, on the other hand, 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: Clustering (OConnor and Herlocker 1999; Han et al. 2001), Naive Bayes (Miyahara and Pazzani 2000; Robles et al. 2003), and Probabilistic models (Breese et al. 1998; Marlin 2003; Hofmann and Puzicha 1999; Hofmann 2004) among others. A good survey of the application of machine learning to collaborative filtering is Marlin (2004).

The main purpose of a collaborative RS involves recommending items to users. Under this formulation, we distinguish two different problems:

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

Although both situations are closely related, this paper deals with the first type.

With respect to the data used for recommendation in a collaborative framework, one can find, on the one hand, a large set of m items, $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$, whose domain can be diverse: books, movies, music, restaurants, web pages, etc. Moreover, there is a large set of n users, $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$. A user U_i can give his opinion about each item using a discrete rating $s, s \in \{1, 2, \ldots, \#r\}$. We can consider these ratings as a high sparse matrix R, of size $n \times m$, where users are represented in the rows and articles in the columns. This matrix is usually sparse, as users usually rate a low number of articles. The value of the matrix, $s_{a,j}$ represents how user U_a has rated item I_j. When a user has not rated a product, the value is 0. For example, Table 1 (left) shows an example of such a matrix.

Collaborative RSs present some well-known problems:

- Sparse rating problem: This problem arises because the number of available ratings previously obtained from users is usually very small compared to the number of ratings needed to achieve reliable predictions. The estimation of new ratings from a small number of examples is thus one of the critical issues in these systems.
- New user problem: When a new user enters the system, no personal ratings are available to him, and no proper recommendations can be made. As recommendations follow from a comparison between the target user and other users, based solely on the accumulation of ratings, if few ratings are available, it may become very difficult to categorize the user's interests.
- New item problem: This is the symmetric counterpart to the new user problem. When a new item is rated by an insubstantial number of users, the RS is unable to recommend it. Hence, a recent item that has not yet obtained many ratings cannot be easily recommended.

Associated with the new item problem and the sparse rating problem is the problem we attempt to solve in the present paper: given an active user, the system requires information on a specific item and people with similar tastes are unable to provide it. In this case, the system will offer an evaluation of the item that will surely be inappropriate. In situations in which no collaborative information exists, content-based approaches have been used in the literature as a possible solution. One approach found to attenuate this problem involves hybrid approaches combining the use of collaborative and content-based technicals. In this case, when no collaborative information is available, the predictions are computed using the ratings given by the active user to those items similar to the target one. The similarity between films is obtained taking into account the features of these items using, for instance, the cosine between the set of features (Ali and van Stam 2004). Another approach is found in Degemmis et al. (2007) in which these authors use similarities between users which rely on their content-based profiles rather than comparing their rating stiles. In Popescul et al. (2001) they use a hybrid Bayesian approach that allows for good recommendations where no collaborative information is available using the EM algorithm. Another bayesian approach is de Campos et al. (2006) in which thanks to the topology of their model, the problem is solved with the use of the collaborative rating to items similar to the target one. Another possible solution found is to use an imputation-boosted collaborative filter (Melville et al. 2001; Su et al. 2008). The aim of the model is to remove the sparseness of the data sets inserting the ratings predicted by a pure content system, i.e., to insert the ratings predicted by the system for every user and every movie that has not been rated. This enables the complete datasets of users' ratings to be used in order to improve predictions. In Table 1 (right), we can see an example of filling a dataset that

Table 1 Left: Data base of user's ratings. Right: imputation-boosted data

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

$\Longrightarrow$

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

contains four users and four movies. Once they have the entire ratings dataset, they build a collaborative RS using the new information available.

3 Using second-hand information

As we have shown, the approaches used to solve the sparsity and the new item problems depend on the availability of content descriptions. In this paper, we present a new possible approach for tackling this problem that can be used in situations in which content description is not available: What we have termed second-hand information. Let us consider the following illustrative example: Imagine that I am Mike and I want to know if I would like the Batman movie. I ask my group of friends (John, Lewis, Eli, Charles and Xavi) if they have seen it (see the modelling of the example in Fig. 1). I ask them because they are my best friends, they know my tastes about movies and we have very similar tastes, and they can therefore tell me whether I will like it or not. John and Lewis have seen the movie, so they can give me their opinion about it. But Eli, Charles and Xavi have not seen it. This is where we apply our idea, i.e., Eli, Charles and Xavi can ask their friends about the Batman movie and then, when we know if they will like the movie, they can give me their opinion about it. For Charles, all his friends have seen the film, and he will therefore obtain good information in this respect. The situation for Eli is the same, so she can obtain a good recommendation from her friends. But only one of Xavi's friends (Henry) has seen the movie. In this case, the information received by Xavi from his friends might not be very accurate. For this reason, Xavi might decide not to give me his opinion, as he considers that is not good enough, i.e., if Xavi thinks the information that he can give me is not quality information, he will not give it to me. Now, once I have all my friends' opinions regarding the Batman movie, I can form a more accurate opinion about whether I will like it or not.

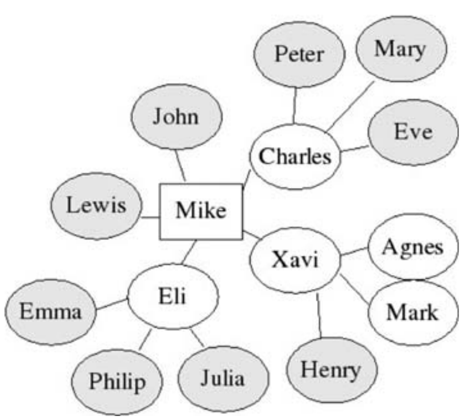

Fig. 1 Obtaining second-hand information

The idea is simple: For those neighbours who did not rate the target item in the past, we obtain new collaborative knowledge using the rating that might be predicted by the system (using the information available from their neighbours).

In order to test whether this idea works we used memory-based RSs. These are based on a two-step process:

1. *Neighbourhood selection*: For every user we obtain a neighbourhood using a metric that obtains the similarity between users. This metric usually depends on the cooccurrency of ratings in the training dataset. Particularly, the top-N users with greater values of similarity are selected as neighbours. We have to note that, for each user, the set of neighbours is fixed and does not depend on the particular item to be recommended.
2. *Computing the predictions*: Given a target item, we can compute the active user's prediction by aggregating in some way the set of ratings given by the neighbours in the past, i.e., $r_a = \texttt{Aggregate}(r_1, \ldots, r_N)$. Note that, since the set of neighbours is fixed, they do not all need to have rated the item previously.

With this recommending philosophy in mind, we can design a recursive[1] algorithm (see Table 2) that takes into account qualified second-hand information when it recommends. In the algorithm, U_i represents a neighbour of the active user A that has been obtained in the *neighbourhood selection* step. The rating prediction (line 9) for an item I by the active user A is made using both the ratings given by the neighbours if they have rated the item (line 3) and those ratings predicted by the system when a neighbour did not rate the item (only if the predicted ratings overcome a quality criteria) (lines 5–6). The function `Qualified` (line 10) returns if a prediction is good enough or not.

Although this approach might help to tackle the sparse rating problem and, in a certain way, the new item problem, its feasibility depends on the qualified second-hand predictions. Thus, even in those cases where we avail of sufficient information, the use of non-qualified information might lead to a worsening of the predictions. Therefore, when there are no available ratings for the target item, this kind of approach might not be helpful. However, since we are using only qualified information, we can expect the performance of the system not to worsen in these situations. Consequently, it might be difficult to recommend an item that has been rated by few users.

Finally, this approach implies a computational cost, as we have to compute the rating predictions for all the neighbours of the active user who have not rated the item. If we assume

[1] To clarify, we show a recursive version of the algorithm, but the implemented version is sequential.

Table 2 Recommending with second-hand information

```
Recommending( Item I, User A, iter it )
Input: I target Item, A active user, it iteration (first or second)
Output: pair<rating,bool> pred
        pred.first is the predicted rating
        pred.second represents the quality of the rating

for (i = 1 to N ) do
  if user U_i rated I
     r_i = rating given by U_i
  else if (it == first)
     p = Recommending(I,U_i,second);
     if (p.second == true) r_i = p.first
     else r_i = unknown
  else r_i = unknown              \\second iteration
pred.first = Aggregate( r_1,...,r_N) \\ using Eq.2 or Eq.5
pred.second = Qualified(pred.first,r_1,...,r_N)
return pred
```

that the similarity weights calculations are computed offline, the final recommendations are obtained in $O(N^2)$, n being the maximum number of neighbours used for recommendation in the system. Nonetheless, and thanks to the offline computations, the approach can be used in online applications, even with a large number of users.

4 Collaborative RSs used

We have used two memory-based RSs: the first, a Pearson correlation-based model and the second, a Probabilistic model based on Bayesian networks (BNs), which we will introduce in Sect. 4.2.

4.1 Weighted average of deviations from the neighbour's mean

Based on the model proposed by Grouplens, weighted average of deviations from the neighbour's mean (Konstan et al. 1997; Herlocker et al. 1999), which hereinafter we call Average model, which is a collaborative RS that uses an algorithm based on neighbourhood.

1. *Neighbourhood selection*: To measure the similarity between users, used as the basis of weights in different collaborative systems, this model is based on the Pearson Correlation Coefficient (PCC), where U is the set of users and U_a two users from U. The PCC can be computed by means of the following formula:

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

where sums over j are applied to those items where users U_a and U_b have ratings, $I(U_a) \cap I(U_b)$ (where $I(U)$ is the set of items rated by the user U in the dataset). If there are no common items between U_a and U_b, then $\mathrm{PCC}(U_a, U_b) = 0$ by default. Furthermore, $\bar{r}_a$ is the average rating value for the user U_a, i.e.:

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

The final value of similarity is computed by applying a correction factor that devalues those PCC values that have been obtained with fewer than 50 items in common (see Herlocker et al. 1999), i.e.:

$$\mathrm{sim}(U_a, U_b) = \mathrm{PCC}(U_a, U_b) \cdot \mathrm{CF},$$

with

$$\mathrm{CF} = \begin{cases} 1 & \text{if } k > 50 \\ \frac{k}{50} & \text{otherwise} \end{cases}$$

k being the number of items in common.

2. *Computing the predictions*: Once we obtain the neighbours for a user, to obtain the predicted rating for an item, the following formula is applied:

$$\mathrm{rate}_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} (r_{u,i} - \bar{r}_u) \cdot \mathrm{sim}_{a,u}}{\sum_{u=1}^{n} \mathrm{sim}_{a,u}}, \qquad (2)$$

where $\mathrm{rate}_{a,i}$ is the rate prediction for the active user A of the item i, n is the number of similar users which have rated the item, $r_{u,i}$ is the rate given by the neighbour U to item i, $\bar{r}_u$ is the average rate of the neighbour and $\mathrm{sim}_{a,u}$ is the similarity measure between the active user and the similar user u.

To finalise the presentation of this model, in this paper we have used a threshold on the number of neighbours used for prediction purposes. As Herlocker et al. (1999) indicates, the use of the best n neighbours performs relatively well without limiting the prediction coverage.

4.2 Probabilistic model: collaborative RS using Bayesian networks

In a certain way, our model is similar to the previously described one, but is based upon a Bayesian formalism in which we consider that all the users are represented as nodes in the BN (see Fig. 2). Particularly, we will include a node A to represent the active user and a subset of nodes in $\mathcal{U}$ to represent those users U_i similar to the active user. On the one hand, the states of each user node, $U_i \in \mathcal{U}$, are in $\{0, 1, 2, \ldots, \#r\}$. Note that state 0 occurs when the user has not rated the item.[2] On the other hand, since A represents the active user's predicted rating, it will take its values in the range of valid ratings, i.e., $\{1, 2, \ldots, \#r\}$. With regard to the aim of this paper, is unnecessary to fully understand the model and we will explain it with little detail (for more details, refer to de Campos et al. 2008).

[2] This is one of the differences from the reference model presented in de Campos et al. (2008), i.e., the inclusion of rating 0 in the performance of the system.

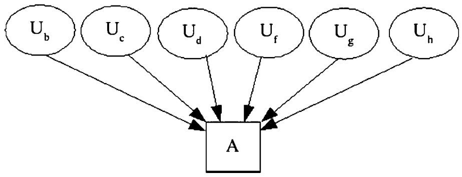

Fig. 2 Probabilistic recommender system topology

1. *Neighbourhood selection*: To measure the similarity between users, we propose a combination of two different but complementary criteria: On one hand, we use Pearson's correlation (see Eq. 1) to capture similarity between users and on the other, we use the overlap degree to penalize spurious correlations

$$\mathrm{sim}(A, U_b) = \mathrm{abs}(\mathrm{PCC}(A, U_b)) \times D(A, U_b).$$

The neighbourhood of the active user A will be obtained using the first n variables in the ranking. Note that we consider similar users with the highest absolute value of PCC. Therefore, both positively (those with similar ratings) and negatively correlated users (those with opposite tastes) will be used to predict the final rating for an active user.[3]

The second criterion attempts to penalize highly correlated neighbours based on very few co-rated items, which have been shown to be bad predictors (Herlocker et al. 1999). The way in which we compute this value varies from the Average model. In particular, we consider that the quality of U_b as the parent of variable A is directly related to the probability of a user U_b rating an item which has been also rated by A. This criterion can be defined by the following expression:

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

where $I(U)$ is the set of items rated by user U in the dataset.

Learning the parameters: One important point to be considered relates to the size of the distributions to be stored in the BN. As the node A is related to a large number of users, we must assess large probability tables. To solve this problem, we propose the use of an additive canonical model (studied in detail in de Campos et al. 2008). 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.

[3] We have evaluated the system with only positive Pearson correlation and we have obtained worst results than using absolute value.

Definition Using the canonical additive model, the set of conditional probability distributions for the active user A can be calculated efficiently in the form:

$$\Pr(A = s|\mathrm{Pa}(A)) = \sum_{U_b \in \mathrm{Pa}(A)} w(U_b = t, A = s),$$

where t is the value that the user U_b takes in the configuration $Pa(A)$ and $w(U_b = t, A = s)$ are weights that measure how the tth value of the user U_b describes the rating sth of the active user A.

The particular way in which the necessary weights are defined is:

$$w(U_b = t, A = s) = \mathrm{RSim}(A, U_b) \cdot \Pr(A = s|U_b = t),$$

RSim being the relative importance of each parent in relation to the active user, defined as

$$\mathrm{RSim}(A, U_b) = \frac{\mathrm{sim}(A, U_b)}{\sum_{U_k \in \mathrm{Pa}(A)} \mathrm{sim}(A, U_k)}.$$

The term $Pr(A = s|U_b = t)$ represents how probable the A rating is with a value s when U_b rated with t. These probabilities are obtained from the dataset of user ratings. The particular way in which we estimate these probabilities will depend on whether U_b rated the target item or not:

- The user rated the target item with t, $U_b = t, t \neq 0$: In order to estimate this probability distribution, we only consider those items which have been rated by both U_b and the active user A, i.e., the set $I(A) \cap I(U_b)$. Particularly,

$$Pr(A = s|U_b = t) = \frac{N(U_b = t, A = s) + 1/\#r}{N(U_b = t) + 1},$$

$N(U_b = t, A = s)$ being the number of times from $I(A) \cap I(U_b)$ which have been rated t by U_b and also s by the active user U_a. In addition, let $N(U_b = t)$ be the number of items in $I(A) \cap I(U_b)$ rated with t by the user U_b.
- The user did not rate the target item, i.e., $U_b = 0$: In this situation we will explore two different options:

V0E: All the ratings for the active user are equally probable, i.e.,

$$\Pr(A = s|U_b = 0) = \frac{1}{r}, 1 \le s \le r. \quad (3)$$

V0A: The contribution to each possible rating will be determined by the a priori probability of the active user, A, i.e.,

$$\Pr(A = s|U_b = 0) = \Pr(A = s), 1 \le s \le r. \quad (4)$$

2. *Computing the predictions*: This model will be used to predict how the active user might rate a target item I. In the BN formalism, this problem is limited to computing the posterior probability distribution for A

given the evidence, i.e., $\Pr(A = s|ev)$ for all valid ratings, i.e., $\{1, 2, \ldots, \#r\}$. The problem now consists of determining what the evidence, ev, is, and how it should be included in the system. We can distinguish between users (parent of A in the BN) who rated the item in the past and those who did not rate it. In the first case, the evidence is the given rating, whereas in the second one, we have instantiate the node to the value 0 (unknown rating). For example, let us assume that $\{U_c, U_d, U_e, U_f\}$ is the set of neighbours of the active user A. Then, if U_c and U_e rated the target item I with 5 and 3, respectively, the evidence set will be $ev = \{u_{c,5}, u_{d,0}, u_{e,3}, u_{f,0}\}$.

We now have all the necessary information to compute the posterior probabilities at node A. This means that, using the advantages of this canonical model, the exact posterior probabilities for the active user (see de Campos et al. 2008) can be computed efficiently as

$$\Pr(A = s|ev) = \sum_{t=0}^{\#r} \sum_{U_b \in \mathrm{Pa}(A)} w(U_b = t, A = s) \cdot \Pr(U_b = t|ev).$$

In our case, as we know whether U_b rated the target item or not, the term $Pr(U_b = t|ev)$ takes only two values. In particular, $Pr(U_b = t|ev) = 1$ if in the evidence U_b rated with t the item, and 0 otherwise. Therefore, these probabilities can be calculated efficiently in the form:

$$\Pr(A = s|ev) = \sum_{U_b \in \mathrm{Pa}(A)} w(U_b = t, A = s). \tag{5}$$

t being the value used to describe the rating of the user U_b to the target item.

Once the a posteriori probabilities have been computed, i.e., when we know $\Pr(A = s|ev)\ \forall s \in \{1, \ldots, \#r\}$, a key issue in the system's performance involves determining how to select the recommended ratings. In the present paper, we will consider two different alternatives for computing a prediction based on the distribution over the ratings (Marlin 2004):

MP: Select the most probable a posteriori rating:

$$\mathrm{rate} = \arg_s \max\{\Pr(A = s|ev)\}.$$

MED: Select the median rating using the a posteriori probability in the form:

$$\mathrm{rate} = \arg\min_s \sum_{i=1}^{s} \Pr(A = i|ev) \geq 0.5.$$

By way of an example, in the case of five possible ratings (ranging from 1 to 5), we obtain the posterior probability distribution $Pr(A = s|ev) = \{0.10, 0.15, 0.30, 0.35, 0.10\}$, i.e., $Pr(A = 1|ev) = 0.10$, $\Pr(A = 2|ev) = 0.15, \ldots, \Pr(A = 5|ev) = 0.10$. The ratings obtained using the different methods are: for MP the rating is 4 as it is the most probable and, for MED is 3 $(0.10 + 0.15 + 0.30 \geq 0.5)$.

4.3 Including second-hand information in the models

To include second-hand information in the models studied, we must change the way in which those models compute the predictions. There is a need to distinguish between the set of neighbours that have rated the movie (R) in the datasets and those for whom we have obtained a rating using second-hand information (NR):

- For the Average model presented in Sect. 4.1, we change Eq. 2 as follows:

$$\mathrm{rate}_{a,i} = \overline{r}_a + \frac{\sum_{u=1}^{R} (r_{u,i} - \overline{r}_u) \cdot \mathrm{sim}_{a,u} + \sum_{u=1}^{\mathrm{NR}} (\hat{r}_{u,i} - \overline{r}_u) \cdot \mathrm{sim}_{a,u}}{\sum_{u=1}^{R} \mathrm{sim}_{a,u} + \sum_{u=1}^{\mathrm{NR}} \mathrm{sim}_{a,u}}$$

where $\hat{r}_{u,i}$ is the rate given by the neighbour U to item I using second-hand information.

- For the probabilistic model presented in Sect. 4.2, we change Eq. 5 as follows:

$$\Pr(A = s|ev) = \sum_{U_b \in \mathrm{Pa}_R(A)} w(U_b = t, A = s) + \sum_{U_b \in \mathrm{Pa}_{\mathrm{NR}}(A)} w(U_b = \hat{t}, A = s).$$

$\hat{t}$ being the value used to describe the rating of the user U_b to the target item using second-hand information.

5 Experimentation

The purpose of this experimentation is to study whether the use of second-hand information might be useful with regard to improve the performance of collaborative RSs. In this section, we will describe the evaluation criteria, the datasets used in the analysis and the particular experimental conditions. We then present and discuss the results regarding predictive accuracy, as well as several computational considerations.

5.1 Evaluation criteria

Our goal is to predict how a given user should rate an item. In this scenario, an individual item will be presented to the users, along with a rating indicating its potential interest. The performance of the model will therefore be evaluated by measuring prediction accuracy. In our paper, the following error measures will be considered, where N is the number of users, p_i the predicted rating and r_i the true rating.

- First, we measure the capacity of the system to predict the correct rating, i.e., the percentage of success of the systems (%S):

$$\%S = \frac{100\sum_{i=1}^{N}[p_i = r_i]}{N}.$$

- Furthermore, we consider the average absolute deviation between a predicted rating and the user's true rating, i.e., the mean absolute error (MAE) defined as

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

- Also, we consider the coverage measure, i.e., the percentage of recommendations made by, at least, one neighbour.

$$\mathrm{coverage} = 100 - \frac{N_0}{N}.$$

Since one of the RS evaluated in this study, the Average model, obtains the ratings on a continuous scale, the percentage of success for this system is inappropriate and will therefore not be shown. Rather, the MAE criteria is obtained for the two models in all the experiments.

5.2 Datasets

Three different datasets were used to evaluate our algorithms. These datasets are available to the public for research purposes:

ML: Database Movielens,[4] containing 1,682 movies and 943 users who provide their ratings to films they have seen, giving rise to 100,000 ratings on a scale of 1–5.
MI-ML: Database Movielens containing 1 million ratings by 6,040 users for 3,900 movies.
JE: Jester Joke dataset.[5] This dataset contains 4.1 million continuous ratings (−10.00 to +10.00) of 100 jokes from 73,496 users. However, all the experiment results from the use of this dataset were scaled down to be equivalent with the other two datasets for easy comparison.

The purpose of this experimentation is to test our approach in conditions in which the predictions are computed, using a small number of ratings, but with available second-hand information. There are two possible situations in which this situation does not arise. First, when many users rated the items (as is the case of the Jester dataset) and, second, when it is very difficult to find similar users who rated the items, due to the sparseness of the data (Movielens datasets). Therefore, and in order to thoroughly explore the situation in which second-hand information is available, we simulated these conditions by randomly removing 50% of the ratings provided by the neighbours of the active user to the target item, i.e., we use the half of the first-hand information. It should be noted that each time we remove one rating, it ceases to be used in all predictions in which it intervened. Hereinafter, we will refer to these datasets as `reduced`.

To understand the impact of the reduction of the datasets, Table 3 shows for each model and each dataset (original and reduced) the number of times that each model predicts a rating, taking into account the past ratings available. The number of used ratings is split into different intervals (less than 6, between 6 and 12, …). As expected, the elimination of ratings increases the number of users in such a way that recommendations are obtained with a low number of ratings, i.e., for example, the numbers for the reduced dataset when neighbours are less than six are always larger than the original because we have removed some of first-hand information.

5.3 Experimental protocols and models' parameters

In the experimentation, we follow a classical protocol in the literature (Breese et al. 1998), where the available ratings for each user are split into an observed set and a held out set. The observed ratings are used for training and the held out ratings are used for testing the performance of

Table 3 Number of past ratings used in the predictions

Dataset	Num. ratings	Probabilistic		Average	
		Original	Reduced	Original	Reduced
ML	<6	4,296	18,865	5,823	18,427
	6…2	5,852	1,131	6,232	1,570
	13…18	5,364	4	4,940	3
	19…24	3,828	0	2,694	0
	>24	660	0	311	0
MI-ML	<6	83,355	187,163	83,355	178,607
	6…12	53,185	5,532	53,185	13,995
	13…18	35,100	15	35,100	108
	19…24	19,071	0	19,071	0
	>24	1,999	0	1,999	0
JE	<6	63	295,029	6,903	288,638
	6…12	2,482	52,400	23,621	58,413
	13…18	9,393	1,013	21,894	1,389
	19…24	179,473	0	156,858	2
	>24	157,031	0	139,166	0

4 http://www.movielens.org.
5 http://www.ieor.berkeley.edu/~goldberg/jester-data/.

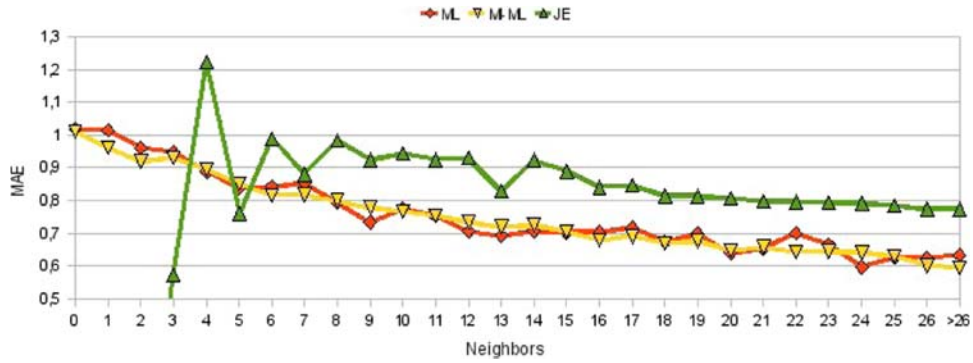

Fig. 3 MAE of the probabilistic model depending on the number of neighbours used to predict the rating

the method. Specifically, we divided the collections into 80% for training and 20% for testing. The final results were obtained by means of cross validation, showing the mean measures obtained across each multiple randomly selected set.

With respect to the parameters of the models, we first wish to discuss the neighbourhood size. On recommending with these models, the use of a small number of neighbours tends to result in greater prediction accuracy (Herlocker et al. 1999). 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. Following the ideas of Herlocker et al. (1999) the neighbourhood size was fixed to 30 in both models, i.e., we consider only the best 30 neighbours for recommendation purposes.

A second parameter can also be discussed. It should be remembered that our aim in this paper is to study whether the inclusion of second-hand information improves the performance of the model. But, as has been pointed out, it seems natural that only those predicted ratings obtained with qualified information will be taken into account. But, what does "qualified information" mean? Our initial hypothesis is that the performance of the collaborative system improves with an increase in the information used for recommending. We have verified this hypothesis experimentally, as shown in Figs. 3 and 4. The abscissa represents the number of similar users who have rated the movie, nr and the ordinate shows the MAE values obtained by the models for each dataset. As can be seen, the behaviour of the systems is similar for the different datasets. It therefore seems reasonable to use a threshold in the number of neighbours who have rated the item as a quality criterion. Specifically, it can be said that a `qualityprediction` exists if it has been obtained using information from at least Q neighbours.

It should be noted that there is a tradeoff between this criterion and the amount of second-hand information used for recommending. Figures 5 and 6 show the percentage of qualified recommendations (coverage) that will be obtained using different number of similar users as the threshold Q. For example for $Q = 5$, i.e., those recommendations made with the use of at least five neighbours, the coverage is about 80% for ML, 90% for MI-ML and 100% for Jester. Looking at these figures, we can also observe a difference between the MovieLens and Jester datasets. It can be seen that the predictions obtained using the Jester dataset are well informed (this is because almost all users rated most of the jokes) where the sparseness of the MovieLens datasets (it is 97.5% sparse) implies that the predictions are obtained with less information. Availing of these data, in our paper we consider as quality ratings those obtained by means of information from at least 40% of neighbours (corresponding with $Q = 12$), because with this threshold we obtain good results in terms of MAE providing a reasonable coverage for both models. Note that the quality value comes into play only in evaluations used to add second-hand information, i.e., if a neighbour of the active user has not rated the movie, the recommended rating for this neighbour is only used if it is obtained by at least Q neighbours (second-hand neighbours).

5.4 Experimental results

5.4.1 Baselines

In this experimentation, our baseline results are those obtained without the use of second-hand information in the two models. Tables 4 and 5 show the results obtained using the Probabilistic-based model and the Average model.[6] We show the results considering the two criteria used to obtain the predicted rating with the Probabilistic model, particularly the most probable (MP) and the median rating (MED). As expected, MP maximizes the $\%S$ and MED minimizes the MAE values. Furthermore, for our probabilistic model, we show the results considering the two different alternatives for distributing the probability mass associated with

[6] Note that in this mode we do not show the success ratio as error measure because the predicted value is not an ordinal value.

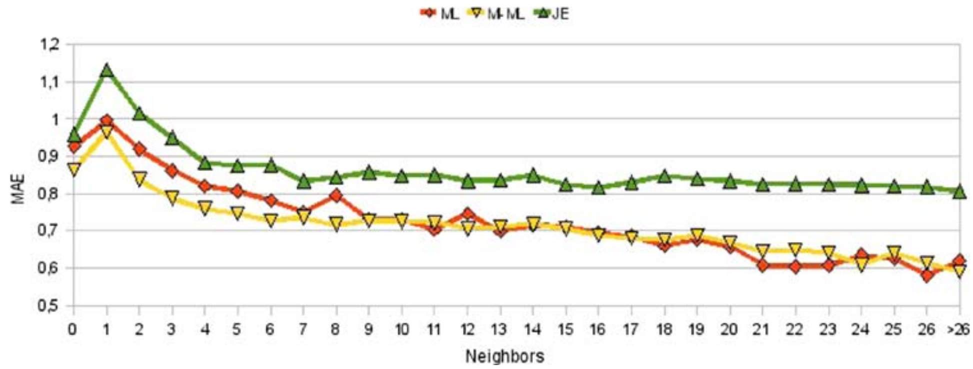

Fig. 4 MAE of the Average model depending on the number of neighbours used to predict the rating

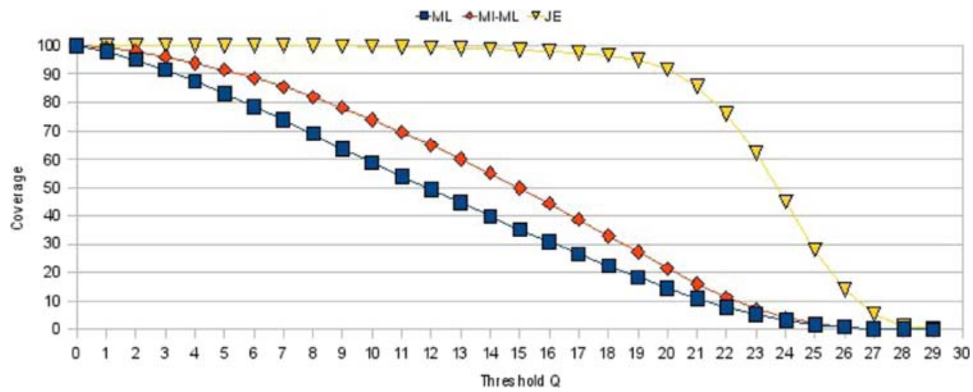

Fig. 5 Accumulated coverage of the Probabilistic model

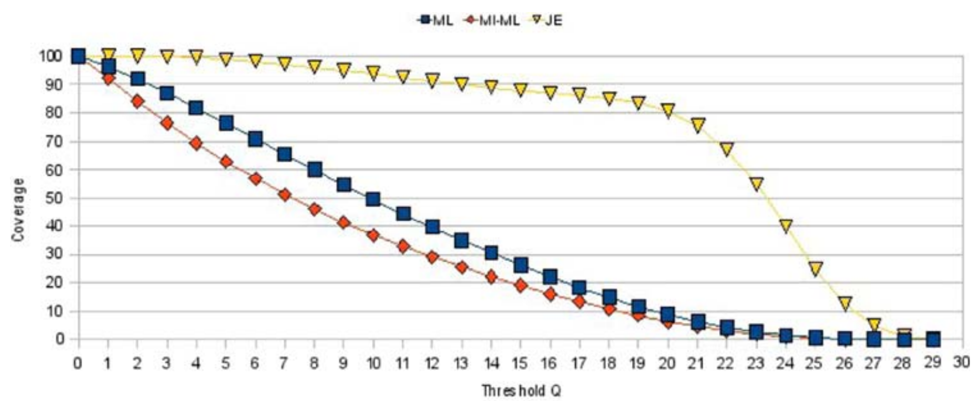

Fig. 6 Acumulated coverage of the Average model

situations in which the neighbours did not rate the target item (state 0): To use the prior probability of the active user (see Eq. 4), denoted as V0A, and to consider all the ratings being equiprobable (see Eq. 3), denoted as V0E. The best results for each dataset are highlighted in bold. Furthermore, Table 6 shows the coverage of the models for each dataset (original and reduced).

These data provide certain conclusions: First, we can see that the results obtained with the Average model are worse than those obtained by means of the Probabilistic model. Moreover, as might be expected, performance of both models presented poorer performance when predicting with less information (reduced datasets). Focussing on the Probabilistic model, significant differences were observed, depending on the method used to distribute the probability mass associated with the lack of information. Thus, with V0A, the performance of the reduced dataset declines by around 7%, whereas with V0E, the performance worsens significantly. Furthermore, the combination of MED + V0A provides the best results in terms of MAE for both datasets. Moreover, the best results in terms of percentage of success, *%S*, were obtained when predicting the most probable rating, but the criterion used to distribute the probability mass associated with the lack of information had a great impact on this metric. With respect of coverage values, we can see how the elimination of ratings cause a

Table 4 Probabilistic models

Dataset	Rate sel.	Original				Reduced			
		V0A		V0E		V0A		V0E	
		%S	MAE	%S	MAE	%S	MAE	%S	MAE
ML	MP	41.62	0.803	**42.38**	0.792	**39.51**	0.839	35.06	1.092
	MED	40.55	0.**755**	36.21	0.799	38.64	**0.791**	26.75	1.010
MI-ML	MP	42.86	0.773	**44.40**	0.737	**39.91**	0.831	28.83	1.425
	MED	41.37	**0.728**	38.37	0.749	38.69	**0.781**	26.19	1.017
JE	MP	45.56	0.803	**46.12**	0.850	**41.93**	0.977	37.77	1.158
	MED	41.25	**0.768**	39.57	0.795	37.66	**0.885**	24.96	1.078

Table 5 Average model

Dataset	Orig. MAE	Reduced MAE
ML	0.762	0.859
MI-ML	0.761	0.838
JE	0.828	0.962

Table 6 Coverage of the models

Dataset	Original		Reduced	
	Prob.	Average	Prob.	Average
ML	97.88	96.41	83.83	87.20
MI-ML	99.20	92.26	64.50	79.41
JE	100	99.98	85.34	87.59

decrease of coverage for each dataset been more clear for the MI-ML dataset. Moreover, we can see that the values obtained for the probabilistic model, in the original datasets, are higher than those obtained for the Average model. This result lead us to say that the neighbourhood obtained for every test user by the probabilistic model are more accurate than those obtained using the Average model.

5.4.2 Using second-hand information

Tables 7 and 8 show the results obtained after inserting qualified second-hand information into both models. In order to facilitate the comparison, we included a + or − symbol to indicate that the results are better or worse than those obtained in the baselines. If we focus on the results obtained with the use of the original dataset, we found that the results are quite similar to the ones obtained without the use of second-hand information. Thus, considering only the best results for the experiments (those in bold face), it can be seen that using Movielens, we obtained slight improvements in terms of MAE, whereas using Jester, worse results were provided. Moreover, in terms of percentage of success, we obtained slightly worse global results. On the other hand, the results reported from the reduced experiment appear to be quite conclusive. Using second-hand information, we might achieve a performance similar to the one obtained with all the available ratings for the active user (original datasets) which may be considered as the optimum. In a certain sense, it seems that we are capable of recovering the predictive capacity of the model. This conclusion is valid for the two collaborative filtering approaches. In Table 9, we can see a demonstration of both situations. If we use the original datasets, the coverage is almost equal to those evaluations without using second-hand information. In contrast, using the reduced datasets, we can see how the increase of coverage is clear respect those evaluations made without using second-hand information.

An in-depth study of both situations will help us to explain this difference in performance. In particular, there are two main situations in which second-hand information might be of no use: First, when the active user avails of sufficient information for recommending, i.e., most of his/her neighbours rated the item. In this case, the second-hand information does not contribute significantly to the computations of the predicted rating. As we discussed, this is the case of the Jester dataset. On the other hand, we may not include sufficient information after consulting our neighbours. This is the case of the original MovieLens datasets.

Table 10 presents the mean number of second-hand ratings added when the number of neighbours who rated the item (NR) ranges from 0 to 12, i.e., the table shows the mean number of second-hand ratings obtained for evaluations made with less than 12 neighbours. The second and third rows show these values when all the available data are used for prediction purposes. The fourth and fifth rows show these values when the artificially reduced dataset is used. We can see how, with the use of all the available ratings, it was difficult to include second-hand ones. On observing the MovieLens datasets, we found that this

Table 7 Inserting second-hand information in the Probabilistic model

Dataset	Rate sel.	Original				Reduced			
		V0A		V0E		V0A		V0E	
		%*S*	MAE	%*S*	MAE	%*S*	MAE	%*S*	MAE
ML	MP	41.61−	0.801+	**42.02**−	0.792	**41.54**+	0.805+	40.73+	0.856+
	MED	40.98+	**0.749**+	38.14+	0.778+	40.46+	**0.758**+	36.56+	0.805+
MI-ML	MP	42.97+	0.765+	**43.73**−	0.744−	**42.62**+	0.776+	42.25+	0.812+
	MED	41.66+	**0.723**+	40.29+	0.731+	41.25+	**0.732**+	38.80+	0.754+
JE	MP	45.79+	0.853−	**45.88**−	0.850	45.55+	0.866+	**46.09**+	0.849+
	MED	41.78+	0.787−	41.74+	**0.785**+	41.27+	0.794+	39.80+	**0.792**+

Table 8 Inserting second-hand information in the Average model

Dataset	Original MAE	Reduced MAE
ML	0.761+	0.792+
MI-ML	0.748+	0.773+
JE	0.836−	0.826+

Table 9 Coverage of the models inserting second-hand information

Dataset	Original		Reduced	
	Prob.	Average	Prob.	Average
ML	97.88	96.43	93.28	91.85
MI-ML	99.21	93.22	94.53	88.37
JE	100	100	100	100

situation holds for rare items (items rated by few people). For example, on average, those items with nr = 10 were rated by 10 and 6% of the users in ML and MI-ML, respectively, and when nr = 5 these items were rated by 5 and 3% of the users in ML and MI-ML, respectively. To the contrary, when the reduced datasets are used, the inclusion of second-hand information becomes more frequent, and as a consequence, the accuracy of the model is improved considerably.

These data lead us to conclude that the inclusion of second-hand information can be beneficial in a collaborative recommending process. When extra information can be included, we might expect better accuracy in the recommendations, whereas the performance is not damaged when this is not the case. This situation holds for the two collaborative approaches considered in the experiment.

5.4.3 Using imputation-based information

We also decided to compare our results with the ones obtained with the use of an imputation-boosted approach. We used the collaborative item-based Naive Bayes classifier described in Su et al. (2008) to fill the complete missing ratings: For each movie I_a, we obtain a classifier in which the class is the movie and we consider as attributes the remaining movies rated in the datasets. For each movie within the attribute set, we learn the weights in respect to the class, using the ratings within the training dataset given by users who have rated both films. Once a classifier for each movie has been created, we obtain the rating prediction for all users who have not seen this movie, using as evidence their ratings of the remaining movies.

Once we have the filled datasets, we use them to predict the ratings in the test sets, by means of our probabilistic model.

Table 11 shows the results obtained when on filling the datasets with the Naive Bayes classifier. The results were obtained using the ML dataset. As can be seen, the results are worse than when second-hand information is used. The main reason for these results might be due to the fact that the datasets are filled a priori, i.e., as we fill all missing ratings, the similarity between users can become distorted and the neighbourhood selection process therefore does not choose the best ones.

6 Discussion

Across the experimentation it has been proven that obtaining new second-hand information might improve the predictions of the systems. An important fact that contributes to the performance, in terms of accuracy, is that the information (in terms of new ratings) must be qualified.[7] This extra information is computed using the set of past ratings available.

The experimental results indicated that there are two possible situations in which the use of second-hand

[7] In order to test this fact we have also included all the ratings but it worsen the performance of the systems.

Table 10 Number of second-hand ratings included in the models

nr	0	1	2	3	4	5	6	7	8	9	10	11	12
ML	0	0.01	0.04	0.16	0.29	0.71	1.22	1.76	2.87	4.2	5.53	6.89	8.49
MI-ML	0.03	0.09	0.32	0.61	1.06	1.78	2.65	3.83	5.06	6.64	8.1	9.55	10.71
ML(R)	10.72	11.22	11.65	12.19	13.11	13.23	13.39	13.71	13.28	14.45	13.43	14	11
MI-ML(R)	17.54	15.47	15.4	15.72	16.04	16.45	16.19	15.93	15.37	14.67	14.16	12.97	13.9

Table 11 Imputation-boosted in the Probabilistic model filling the ratings using a NB classifier

	%S	MAE
MP	40.65	0.889
MED	39.58	0.786

information does not contribute to the rating prediction: First, when new information cannot be obtained from the database of ratings, as occurs with rare items. In this situation, content information (if available) can be used in order to make the predictions, as with hybrid approaches. Furthermore, the use of second-hand information will not help if sufficient first-hand ratings already exist. Nevertheless, in both situations, the performance of the system is not worsened when second-hand information is resorted to. We therefore consider the situations in which our approach might be useful. The following are two possible ones: On one hand, we consider the case in which the target item is neither rare nor highly frequent (this could be the common one in many applications). It might therefore be interesting to look in the database of ratings in search of extra information. Second, the approach can be useful in an online store (such as Amazon or a movie-based application) where new products frequently appear. In these stores, the users start to rate after the new item is included. Therefore, at the beginning (e.g., a few weeks) it is possible that, given a user, few of his/her neighbours have rated it. In this particular situation the information provided by second-hand neighbours is welcome (our experiments with the reduced dataset reinforce this idea). Therefore, as a final conclusion, as our approach is efficient (in time) and effective (it could improve the predictions), we believe that implementation thereof in real RSs can be beneficial.

7 Conclusions

We have presented a novel idea for application in collaborative RSs in order to improve the predictions of the system by increasing the available information in the datasets using the predictions made by the system. To test it, we used two RSs: A BN-based RS and a neighbourhood-based rating prediction.

We have proved that if we introduce quality second-hand information in the systems, their recommendations can also be improved. This situation is particularly beneficial when the amount of second-hand ratings included is large. Rather, when we use the original datasets, in predictions executed with little information, introducing quality ratings produces a low or almost null increase in information for such predictions. We have demonstrated that this is due to the existence of "rare" items rated by few users.

As future work, we will consider:

- Using additional databases as NetFlix to test our proposal.
- Performing a more exhaustive study of those evaluations in which few neighbours are used for the recommendation, in order to achieve better performance with the original datasets.
- Finding new methods to predict the rating as a mixing of the MP and MED criterion.
- Testing different methods to obtain the neighbourhood.
- Finding alternative criteria to define quality, such as using the final value of probability to assess this.
- Studying new ways to calculate the weights in order to improve overall system performance.
- Incorporating content information to the models to improve the new-items situations.

Acknowledgments This study has been jointly supported by the Spanish Ministerio de Educación y Ciencia under the projects TIN2005-02516 and TIN2008-06566-C04-01, and the research program Consolider Ingenio 2010 under the project CSD2007-00018.

References

Ali K, van Stam W (2004) Tivo: making show recommendations using a distributed collaborative filtering architecture. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 394–401. doi:10.1145/1014052.1014097

Belkin NJ, Croft WB (1992) Information filtering and information retrieval: two sides of the same coin? Commun ACM 35(12):29–38. doi:10.1145/138859.138861

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: 14th conference on uncertainty in artificial intelligence, pp 43–52. http://citeseer.ist.psu.edu/breese98empirical.html

Burke R (2002) Hybrid recommender systems: survey and experiments. User Model User Adapted Interact 12(4):331–370

de Campos LM, Fernández-Luna JM, Huete JF (2006) A Bayesian network approach to hybrid recommending systems. In: Eleventh international conference of information processing and management of uncertainty in knowledge-based systems. Paris, France, pp 2158–2165

de Campos LM, Fernández-Luna JM, Huete JF (2008) A collaborative recommender system based on probabilistic inference from fuzzy observations. Fuzzy Sets Syst. 159(12):1554–1576. doi: 10.1016/j.fss.2008.01.016

Degemmis M, Lops P, Semeraro G (2007) A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. User Model User Adapted Interact 17(3):217–255. doi:10.1007/s11257-006-9023-4

Han S, Chee S, Han J, Wang K (2001) Rectree: an efficient collaborative filtering method. In: Lecture Notes in Computer Science: Data Warehousing and Knowledge Discovery, pp 141–151

Herlocker JL, Konstan JA, Borchers A, Riedl J (1999) An algorithmic framework for performing collaborative filtering. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 230–237. doi:10.1145/312624.312682

Hofmann T (2004) Latent semantic models for collaborative ltering. ACM Trans Inf Syst 22(1):89–115

Hofmann T, Puzicha J (1998) Latent class models for collaborative filtering. In: 16th international joint conference on artifical intelligence. Stockholm, Sweden, pp 688–693

Kangas S (2002) Collaborative filtering and recommendation systems. In: VTT information technology

Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J (1997) GroupLens: applying collaborative filtering to Usenet news. Commun ACM 40(3):77–87. http://citeseer.ist.psu.edu/konstan97grouplens.html

Marlin B (2004) Collaborative filtering: a machine learning perspective. University of Toronto

Marlin B (2003) Modeling user rating profiles for collaborative filtering. In: NIPS*17. MIT Press, Cambridge

Melville P, Mooney RJ, Nagarajan R (2001) Content-boosted collaborative filtering. In: Proceedings of the 2001 SIGIR workshop on recommender systems

Miyahara K, Pazzani MJ (2000) Collaborative filtering with the simple bayesian classifier. In: Pacific Rim International conference on artificial intelligence, pp 679–689. http://citeseer.ist.psu.edu/miyahara00collaborative.html

OConnor M, Herlocker J (1999) Clustering items for collaborative ltering. In: ACM SIGIR 99 workshop on recommender systems: algorithms and evaluation

Popescul A, Ungar L, Pennock D, Lawrence S (2001) Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In: 17th conference on uncertainty in artificial intelligence. Seattle, Washington, pp 437–444. http://citeseer.ist.psu.edu/popescul01probabilistic.html

Resnick P, Varian HR (1997) Recommender systems. Commun ACM 40(3):56–58. doi:10.1145/245108.245121

Robles V, Larrañaga P, Peña J, Marbán O, Crespo J, Pérez M (2003) Collaborative filtering using interval estimation naive bayes. In: Lecture Notes in Artificial Intelligence: Advances in Web Intelligence, pp 46–53

Su X, Khoshgoftaar TM, Zhu X, Greiner R (2008) Imputation-boosted collaborative filtering using machine learning classifiers. In: SAC '08: Proceedings of the 2008 ACM symposium on applied computing. ACM, New York, pp 949–950. doi:10.1145/1363686.1363903