Expert Systems with Applications 41 (2014) 2065–2073

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Facing the cold start problem in recommender systems

Blerina Lika, Kostas Kolomvatsos *, Stathes Hadjiefthymiades

Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece

ARTICLE INFO

Keywords:
Recommender systems
Cold start problem

ABSTRACT

A recommender system (RS) aims to provide personalized recommendations to users for specific items (e.g., music, books). Popular techniques involve content-based (CB) models and collaborative filtering (CF) approaches. In this paper, we deal with a very important problem in RSs: The cold start problem. This problem is related to recommendations for novel users or new items. In case of new users, the system does not have information about their preferences in order to make recommendations. We propose a model where widely known classification algorithms in combination with similarity techniques and prediction mechanisms provide the necessary means for retrieving recommendations. The proposed approach incorporates classification methods in a pure CF system while the use of demographic data help for the identification of other users with similar behavior. Our experiments show the performance of the proposed system through a large number of experiments. We adopt the widely known dataset provided by the GroupLens research group. We reveal the advantages of the proposed solution by providing satisfactory numerical results in different experimental scenarios.

1. Introduction

Recommender systems (RSs) technology currently is in use in many application domains. RSs can suggest items of interest to users based on their preferences. Such preferences could be retrieved either explicitly or implicitly. In general, recommendations are based on models built from item characteristics or users' social environment. For example, recommendations could be based on preferences of other users having similar characteristics (e.g., age, occupation). The recommendation result is the outcome of a complex process that combines the attributes of items and information about users. Recommendation algorithms try, through intelligent techniques, to identify possible connections between items and users and give the most efficient results. The final aim is the maximization of the quality of recommendation (QoR). As QoR could be defined the value of the matching between a specific item and a specific user.

In literature, one can find the following techniques adopted in RSs: (a) *Collaborative filtering (CF) methods* (Tso-Sutter, Marinho, & Schmidt-Thieme, 2008; Das et al., 2007; Sarvar et al., 2001; Schafer, Frankowski, Herlocker, & Sen, 2007; Jambor & Wang, 2010; Jin, Si, & Zhai, 2006; Wang, de Vries, & Reinders, 2006; Rocchio, 1971; Khabbaz & Lakshmanan, 2011; Popescul, Ungar, Pennock, & Lawrence, 2001) and (b) *Content-based (CB) methods* (Pazzani & Billsus, 2007; Lops, de Gemmis, & Semeraro, 2011; De Gemmis, Lops, & Semeraro, 2007; Middleton, Shadbolt, & De Roure, 2004; Billsus & Pazzani, 2000; Mooney & Roy, 2000). CF systems try to retrieve the final recommendation result through community preferences. Usually, in such systems demographics or user attributes are neglected (Schein, Popescul, Ungar, & Pennock, 2002). More specifically, CF approaches recommend items to a target user based on given ratings by other users in the community. Many algorithms have been proposed for the calculation of similarities between users or items. The selection of the algorithm plays an important role to the final QoR. CB systems try to match user profiles against items description. CB approaches require ratings made by the user herself in contrast to CF models that cannot derive an efficient result without the ratings of other users. Additionally, *Hybrid methods* have been proposed (Popescul et al., 2001; Schein et al., 2002) in order to cover the disadvantages of CF and CB models. These methods combine both techniques in order to provide a more efficient result. Many promising algorithms were presented in the above categories, however, some issues are still open.

One of the most known problems in RSs is the cold start problem. The cold start problem is related to the sparsity of information (i.e., for users and items) available in the recommendation algorithm. The provision of a high QoR in cold start situations is a key challenge in RSs (Park & Chu, 2009). Three types of cold start problems could be identified: (a) recommendations for new users, (b) recommendations for new items, and (c) recommendations on new items for new users. Researchers try to overcome the discussed problem, however, they are interested mainly in item side cold start problems (Zhang, Liu, Zhang, & Zhou, 2010). In this paper, we focus on solving the user side cold start problem. We consider the scenario where a new user

* Corresponding author. Tel.: +30 7275127.
E-mail addresses: mop10319@di.uoa.gr (B. Lika), kostasks@di.uoa.gr (K. Kolomvatsos), shadj@di.uoa.gr (S. Hadjiefthymiades).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.09.005

asks for recommendations and no data are available for her preferences. Such data are related to ratings for items. Ratings are very important as they show the preferences of a specific user. Additionally, no historical data are present. We propose an algorithm which results the final outcome through three phases. The first phase is responsible to provide means for the classification of the new user in a specific group. For the classification, we adopt efficient techniques like the C4.5 algorithm (Kotsiantis, 2007) and the Naive Bayes algorithm (Zhang, 2004). In the second phase, the algorithm utilizes an intelligent technique for finding the 'neighbours' of the new user. We examine important characteristics of the user and try to find other users inside the group that best match to her. In the third phase, the final outcome is calculated. This is done adopting prediction techniques for estimating the ratings of the new user. In comparison with research efforts found in the literature, our work has the following differences. Our model:

- handles the new user cold start problem,
- does not require any a priori probability to be known like efforts adopting probabilistic models,
- does not require any interview process,
- does not depend on any complex calculations,
- involves semantic similarity metrics in the calculation process.

The structure of the paper is as follows. Section 2 describes important research efforts in the domain of RSs and the cold start problem. Section 3 gives a high level description of the proposed system while Section 4 presents in detail the key components of our RS. Section 5 is devoted to the presentation of evaluation metrics and the description of our experimental results. Finally, Section 6 concludes the paper.

2. Related work

The CF methods are categorized to (Khabbaz & Lakshmanan, 2011): (i) user-based, (ii) item-based, (iii) model-based, and, (iv) fusion-based approaches. In the user-based approaches (Herlocker, Konstan, Borchers, & Riedl, 1999), a similarity matrix is adopted to store the ratings of each user for every item. The main problem is when missing values are present. The item-based methods (Deshpande, 2004; Sarvar et al., 2001; Wang et al., 2006) adopt pairwise item similarities which are more reliable than user similarities, thus, resulting in higher QoR. The model-based methods (Das et al., 2007; Canny, 2002; Jin et al., 2006) exploit the sparsity of data in the similarity matrix. Training examples are used to generate the appropriate model parameters. Based on such parameters, missing values could be substituted. However, tuning a significant number of parameters has prevented these methods from wide adoption (Jambor & Wang, 2010). The fusion-based methods (Tso-Sutter et al., 2008; Zhang et al., 2010; Oku & Hattori, 2011) adopt information fusion techniques for building the final items list.

As mentioned, CB systems try to match user profiles against items description. Various techniques have been used in CB models like keyword-based models (Asnicar & Tasso, 1997; Chen & Sycara, 1998; Mladenic, 1999; Moukas, 1997), semantic techniques (Basile, de Gemmis, Gentile, Iaquinta, & Lops, 2007; De Gemmis et al., 2007; Eirinaki, Vazirgiannis, & Varlamis, 2003; Magnini & Strapparava, 2001; Middleton et al., 2004) or probabilistic models (Billsus & Pazzani, 1999; Billsus & Pazzani, 2000; Mooney & Roy, 2000; Pazzani & Billsus, 1997). Keyword-based models handle every document as a vector where each dimension describes a specific term. Weights are used to define the association between documents and terms (i.e., user profiles and item characteristics). Semantic techniques provide means for reasoning in the recommendation mechanisms. Ontologies play an important role to that, however, one can identify the problem of heterogeneity. Different item providers could utilize different ontologies, thus, the reasoning process becomes very hard. Probabilistic methods yield posterior probabilities by analysing historical data and based on a priori probabilities. Such probabilities are related to the relationship between documents and terms. Usually, the Bayes rule is the key methodology for the calculation of such probabilities while the Naive Bayes classifier is recognized as the method with the best performance (Lewis & Ringuette, 1994).

The adoption of CB systems has a number of advantages and disadvantages (Lops et al., 2011). These approaches require ratings made by the user herself in contrast to CF models that cannot derive an result without other users ratings. However, in CB systems the cold start problem is intense as ratings are not available for new users. CB models depend on the performance of content analysis methodology they adopt. Explanations on the final result could be given by terms of items descriptions and users profiles something that cannot be done in CF approaches. Additionally, new items are handled easier as the recommendation is based on their descriptions even if ratings are not present yet. The performance of the matching process of the item descriptions with the user profiles is also a critical issue.

A number of research efforts deal with the cold start problem. The combination of collaborative data and content is proposed as a solution to the discussed problem (Popescul et al., 2001; Schein et al., 2002). Such models incorporate three data sources: users, items and item contents. The influence of collaboration data with content emerges naturally from the given data sources by adopting a probabilistic model. Six techniques that CF systems can use to learn about new users are presented in Al Mamunur et al. (2002). These techniques select a sequence of items to present to every new user. In Massa and Bhattacharjee (2004), the authors try to assert the cold start problem through a trust-aware system that takes into account the 'web of trust' provided by every user. The proposed model involves trust propagation between users and inference on the weights of unknown users. A recommendation algorithm based on social tags is proposed in Zhang et al. (2010). The algorithm provides personalized recommendations especially when the assigned tags belong to diverse topics. In Lam, Vu, Le, and Duong (2008), an hybrid approach is discussed. The proposed model utilizes a combination of the CF approach with the CB. Two probabilistic aspect models using pure CF try to handle the new user problem. The use of association rules is the proposed solution in Shaw, Xu, and Geva (2010). Through such rules, the authors try to expand the user profile and, thus, avoid the cold start problem. The performance is improved using non-redundant rule sets. However, complete rule enumeration is often intractable for datasets with a very large number of multi-valued attributes. In Zhou, Yang, and Zha (2011), the authors present functional matrix factorization (fMF). fMF constructs a decision tree for the initial interview (each node being an interview question) enabling the RS to query the user adaptively. Hence, the interview phase could be 'alleviated' thus improving the performance of the model. In Park and Chu (2009), the authors propose predictive feature-based regression models that leverage all the available information of users and items to tackle the cold-start problem. Finally, in Golbandi, Koren, and Lempel (2011), a model for the profiling of new users is discussed. The proposed model is a kind of an interview that elicits the opinion of users about items. The model involves an adaptation scheme on the users' answers in order to provide a more efficient result.

3. The proposed model

The proposed model alleviates the user cold start problem of CF. The main operational aspects are depicted in Fig. 1. The process of

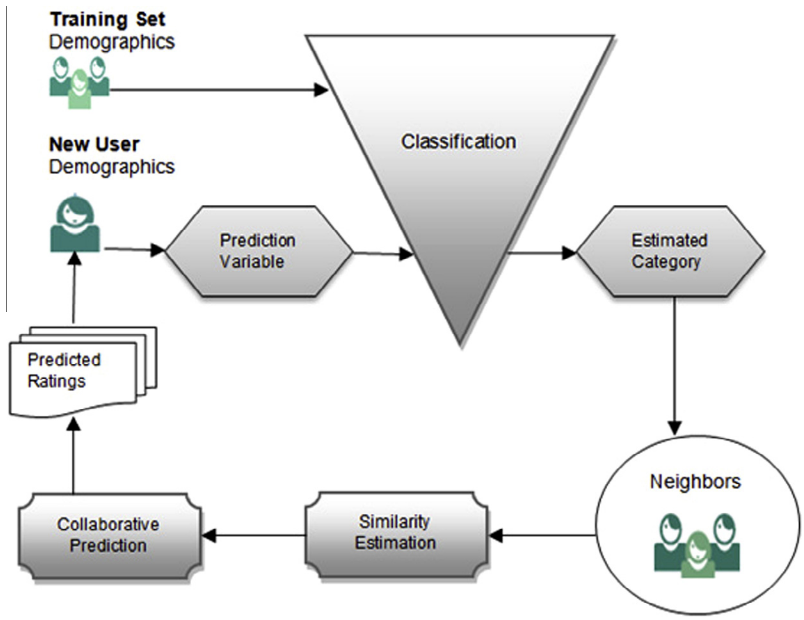

Fig. 1. The architecture of the proposed system.

predicting item ratings for a new user involves three phases. Let the set of current users in the system be $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and $\mathcal{N} = \{n_1, n_2, \ldots, n_n\}$ be the set of the new users. Moreover, a set $\mathcal{I} = \{i_1, i_2, \ldots, i_k\}$ of items is available.

At first, we build a model based on demographic data $\mathcal{D} = \{d_1, d_2, \ldots, d_l\}$ ($\mathcal{D}$ is defined by developers) and users' preferences. We name this step '*Classification*'. The idea is that people with a common background are much likely to have similar preferences. The classification component implements a model on the basis of a training set that contains instances of the whole data store. Instances include variables related to $\mathcal{D}$. Then, we use the generated model to map a new observation in the appropriate category C. C belongs in the set of categories $\mathcal{C} = \{C_1, C_2, \ldots, C_b\}$ ($\mathcal{C}$ is defined by developers). In Fig. 1, we show two key factors: (a) *the prediction variable* $\tilde{V}$, and (b) *the estimated category* $\tilde{C}$. For each new observation $O_j,\ j = 1, 2, \ldots$, we set a class attribute that represents $\tilde{V}$. Values of this class are the possible categories $\bigcup c_j \subseteq \mathcal{C}$ for each new O_j. One of these categories c_j is the output of the model and the corresponding category $\tilde{C}$ for every $n \in \mathcal{N}$. The goal is to find a neighbourhood $\mathcal{NG} = \bigcup u_j$ ($\mathcal{NG} \subseteq \mathcal{U}$) for $n \in \mathcal{N}$. The neighbours in $\mathcal{NG}$ are users that belong to the same category as the model predicts.

Second, after the selection of $\mathcal{NG}$, we calculate the similarity between $n \in \mathcal{N}$ and each of the neighbours $u_j \in \mathcal{NG},\ j = 1, 2, \ldots, |\mathcal{NG}|$ through a weighted average of demographic data. We name this step as '*Similarity Estimation*'. In this phase, we incorporate a similarity function that combines similarity weights from different $d_j \in \mathcal{D},\ j = 1, 2, \ldots, |\mathcal{D}|$. In case of numeric data, we use a particular exponential function as described in the following section. For literal attributes, we use a semantic similarity measure (Wu and Palmer Metric (Wu & Palmer, 1994)). However, in case of binary literal attributes, we take as similarity result boolean values (true or false).

Finally, we make predictions combining the similarity measure and neighbours ratings. We name this step as '*Collaborative Prediction*'. This component implements a function that makes a prediction for an item $i \in \mathcal{I}$. The prediction is derived by a weighted average of each u_j ratings. More specifically, we combine the similarity weights calculated in the previous phase with the ratings of neighbours for the possible recommended item.

4. Model analysis

4.1. User classification

As discussed, through the use of classifications algorithms, we are able to produce $\mathcal{C}$ based on the data related to the set $\mathcal{U}$. In order to have the final $\mathcal{C}$, we apply binary classifiers while the application of multi class classifiers gives us the opportunity to have multiple classes in the results. In the proposed system, we combine a binary classifier with the model one-against-all (OvA) (Milgram, Cheriet, & Sabourin, 2006) for achieving multi-class classification. At first, we train the system with the set $\mathcal{U}$ and, accordingly, we predict the category C_j for the user n_j. In Algorithm 1, we provide the OvA algorithm.

For predicting the new category C_j of a new instance, we apply the generated model. The new category satisfies the following equation:

$$\hat{y} = \arg\max_{1 \leqslant k \leqslant K} f_k(x) \quad (1)$$

In this work, we adopt as binary classifier the C4.5 (Kotsiantis, 2007) and the Naive Bayes algorithm (Zhang, 2004). In Fig. 1, we see that the result of the discussed process is the estimated category $\tilde{C}$.

Algorithm 1. OvA

Input: L, Instances X, Labels Y! L: Training Algorithm
Output: Classifiers $f_k,\ k = 1, 2, \ldots, K$
Begin
  forall $k \in \{1, 2, \ldots, K\}$ **do**
    if $y_i = k$ **then**
      $y'_i = 1$
    else
      $y'_i = 0$
    end if
    $f_k = L(X, y')$
  end for
End

4.2. Users similarity

After calculating $\tilde{C}$, we should proceed with grouping users. Our aim is to find the neighbourhood of each user $n_j \in \mathcal{N}$. The algorithm for finding the neighbors of n_j is depicted by Algorithm 2. The proposed algorithm aims to match $\tilde{C}_{n_j}$ against $\tilde{C}_{u_j}$, where $\tilde{C}_{n_j}$ is the category of the new user $n_j \in \mathcal{N}$ and $\tilde{C}_{u_j}$ is the category of the user $u_j \in \mathcal{U}$. The result of the discussed algorithm is the set of neighbours $\mathcal{NG}$.

Algorithm 2. Neighborhood Calculation

```
Input: U, N
Output: NG for each new user
Begin
  Define options: OvA
  Set Class Index
  Build C4.5 Tree
  Build the MultiClass Classifier
  for all n_j ∈ N do
    NG = null
    Find C̃_{n_j}
    for all u_j ∈ U do
      Find C̃_{u_j}
      if C̃_{n_j} = C̃_{u_j} then
        NG.add(u_j)
      end if
    end for
  end for
End
```

The next step is to calculate the similarity between new users and users in the $\mathcal{NG}$ set. Therefore, the final prediction is based on ratings of the nearest neighbors. The similarity results concern the demographic attributes as defined by $\mathcal{D}$. The final similarity degree is calculated through the following equation:

$$sim(n,u) = \frac{\sum_{j=1}^{l} SF_j \cdot w_j}{\sum_{j=1}^{l} w_j} \tag{2}$$

where SF_j is the similarity value of the jth attribute and w_j is the corresponding weight. Through this equation, we provide a framework where the developer can focus on specific demographic data. For example, let us consider $\mathcal{D} = \{d_1 = age, d_2 = occupation, d_3 = gender\}$. The discussed set $\mathcal{D}$ can be easily extended. We can focus on age, if we define $w_1 = 0.5, w_2 = 0.25, w_3 = 0.25$. When $w_j = 1.0$, the calculation process is fully based on the j^{th} attribute.

For each attribute d_j, we define a similarity function $SF(at_1, at_2) \in [0,1]$ that gives the results every similarity value SF_j. The terms at_1 and at_2 are the attribute values for a pair of users under consideration. We consider two attribute categories: (a) numeric, (b) literal. For numerical values, we adopt a function $SF : \mathbb{R}^+ X \mathbb{R}^+ \rightarrow [0,1]$ while for literal values, we adopt semantic similarity techniques. In the discussed example, we consider a SF for defining the weight of the age $w_a \in [0,1]$ as follows:

$$w_a = \begin{cases} \left(1 - \frac{|Diff|}{Diff_{max}}\right)^{\omega} & \text{if } |Diff| \leqslant Diff_{max} \\ 0 & \text{if } |Diff| > Diff_{max} \end{cases} \tag{3}$$

where $Diff$ is the difference in age between two users and $Diff_{max}$ is a maximum difference (defined by developers). The ω parameter is a policy factor. If the developer wants to have an increased value of the weight w_a even for large $Diff$ values, then she will adopt a very small ω value (smaller than 1). The opposite stands when ω is large. For literal attribute values, we adopt the known Wu–Palmer semantic similarity metric (Wu & Palmer, 1994). Wu–Palmer metric adopts the known Least Common Subsummer (LCS) technique. This technique finds the common node of the two examined issues in the Wordnet taxonomy (<http://wordnet.princeton.edu/>). Finally, in the case of binary literal attribute (i.e., gender) or binary numerical attribute values, we consider boolean similarity values (true or false). Hence, $SF(at_1, at_2) = 1$ when $at_1 = at_2$ and $SF(at_1, at_2) = 0$ when $at_1 \neq at_2$.

4.3. Ratings prediction

The final phase is to produce predictions for new users. For each user $n_j \in \mathcal{N}$, the model should provide predicted ratings for every item $i_b \in \mathcal{I}$. Every predicted rating $R_{n_j,i_b} \in \mathbb{R}^+$ is a weighted sum of ratings made by the users in $\mathcal{NG}$. Therefore, the following equation holds true:

$$R_{n_j,i_b} = \frac{\sum_{u \in \mathcal{NG}} sim(n_j,u) \cdot r_{u,i_b}}{\sum_{u \in \mathcal{NG}} sim(n_j,u)} \tag{4}$$

where r_{u,i_b} is the rating of the user u for the item i_b. Based on the above approach, we aim to enhance ratings that are made by users having large similarity degree with every new user. This is as expected as users having in common a lot of characteristics probably they will have similar item preferences.

5. Performance assessment

We report on the performance of the proposed model. We define certain performance metrics and, then, present our results. Our aim is to quantify the performance of the proposed model concerning the prediction accuracy and compare the results obtained by using different classification algorithms.

5.1. Evaluation metrics

We adopt widely used metrics for prediction accuracy. The first metric is the *Mean Absolute Error (MAE)*. MAE is defined in Eq. (5). In this equation, $p_{u,i}$ defines the prediction for user u and for item i while $r_{u,i}$ symbolizes the actual rating. Finally, with K, we symbolize the number of items under evaluation.

$$MAE = \frac{1}{K} \sum_{u,i} |p_{u,i} - r_{u,i}| \tag{5}$$

Another important metric is the *Root Mean Square Error (RMSE)* defined by Eq. (6).

$$RMSE = \sqrt{\frac{1}{K} \sum_{u,i} (p_{u,i} - r_{u,i})^2} \tag{6}$$

Both metrics are widely used in evaluating recommender systems with respect to prediction accuracy.

5.2. Experimental evaluation

We run a number of experiments for a specific dataset. The dataset is retrieved by the GroupLens research team (<http://www.grouplens.org/>). GroupLens provides the MovieLens dataset containing one million ratings for 4000 movies defined by 6000 users. From the set of users, we choose a number as the registered users in the system and the rest are considered as new users. We start from 100 registered users and, in different scenarios, we increase the number till 5000 users. Through this approach, we try to find out how the system behaves for different numbers of

Table 1
Experimental parameters.

Parameter	Values
Algorithm	$C^2 4.5, C^M 4.5$, NB, RCA
w_j	$w_j \in [0,1], \sum_{j=1}^{3} w_j = 1$
ω	0.8

Table 2
Experimental scenarios.

Scenarios	Weights
Scenario 1	$w_1 = 0.6, w_2 = 0.3, w_3 = 0.1$
Scenario 2	$w_1 = 0.3, w_2 = 0.6, w_3 = 0.1$
Scenario 3	$w_1 = 0.3, w_2 = 0.1, w_3 = 0.6$
Scenario 4	$w_1 = 0.33, w_2 = 0.34, w_3 = 0.33$

registered users. Ratings are between 1 (minimum value) and 5 (maximum value). All ratings are integer values. For each user, we take the identification number and her demographic data $\mathcal{D} = \{d_1, d_2, d_3\} = \{age, occupation, gender\}$. Moreover, we consider that $\mathcal{C} = \{C_1, C_2, C_3, C_4\} = \{fun, intellectual, adventurous, romantic\}$. Both lists $\mathcal{D}$ and $\mathcal{C}$ could be easily extended.

In our experiments, we adopt two classifiers: the C4.5 algorithm and the Naive Bayes (NB) approach. Additionally, we adopt a technique that randomly classifies each user in $\mathcal{C}$. This methodology is named Random Classification Algorithm (RCA). For the C4.5 case, we examine a scenario where only two classes are used for the classification of each user (depicted by $C^2 4.5$) and a scenario where multiple classes are considered in the classification process (depicted by $C^M 4.5$). We compare results taken by the three discussed models (i.e., C4.5, NB and RCA). We examine a number of scenarios defined by the values of weights for each d_j. In Table 1, we give a short description of our parameters.

5.3. Experimental scenarios

In Table 2, we depict our experimental scenarios. These scenarios are defined through the w_j values. Every combination deals with the parameter on which the proposed system pays more attention. For instance, in Scenario 1, the system focuses primarily on the “age” parameter in order to issue the required

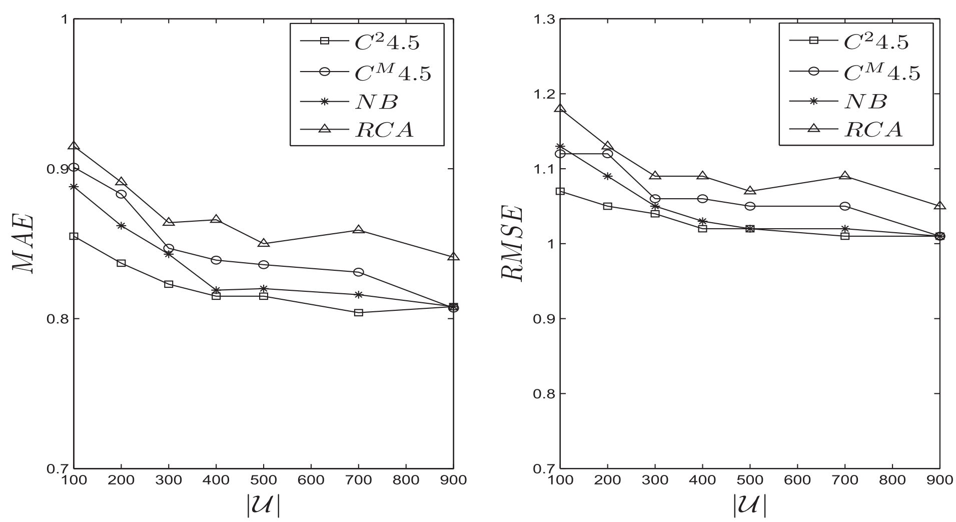

Fig. 2. Results for Scenario 1.

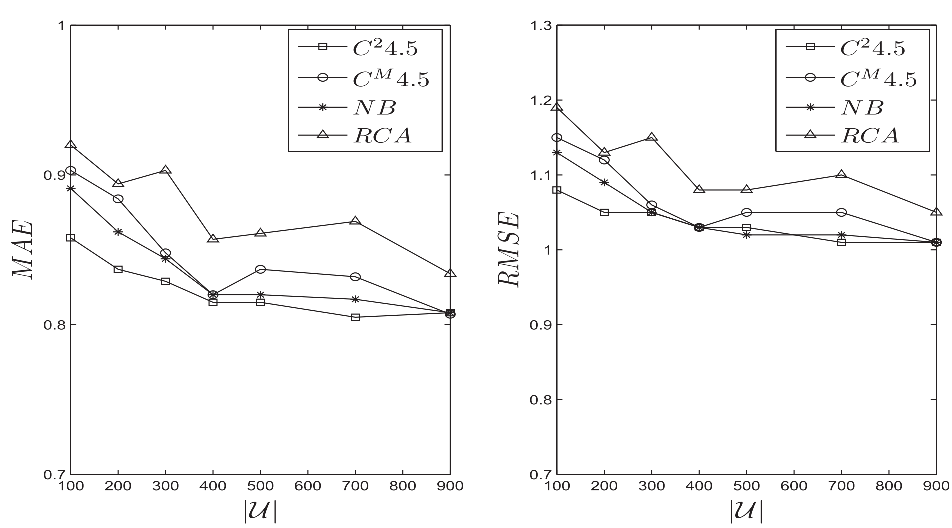

Fig. 3. Results for Scenario 2.

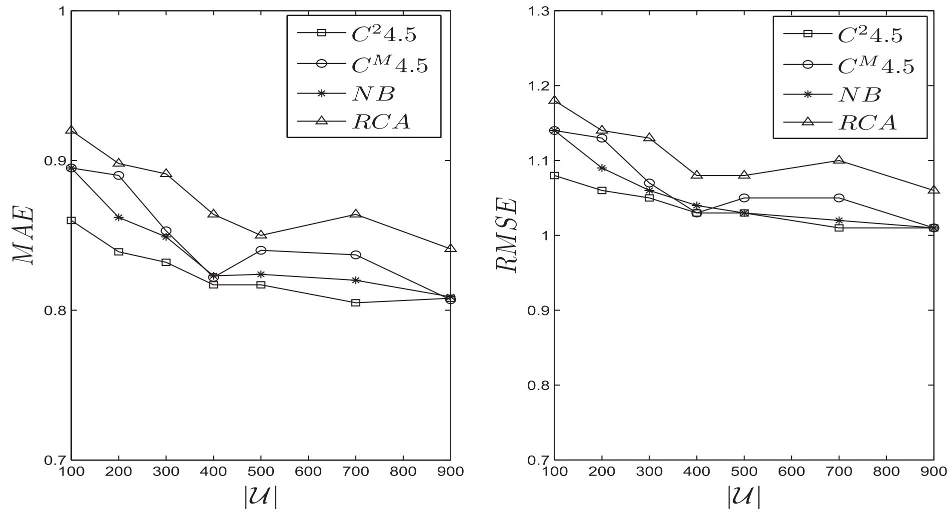

Fig. 4. Results for Scenario 3.

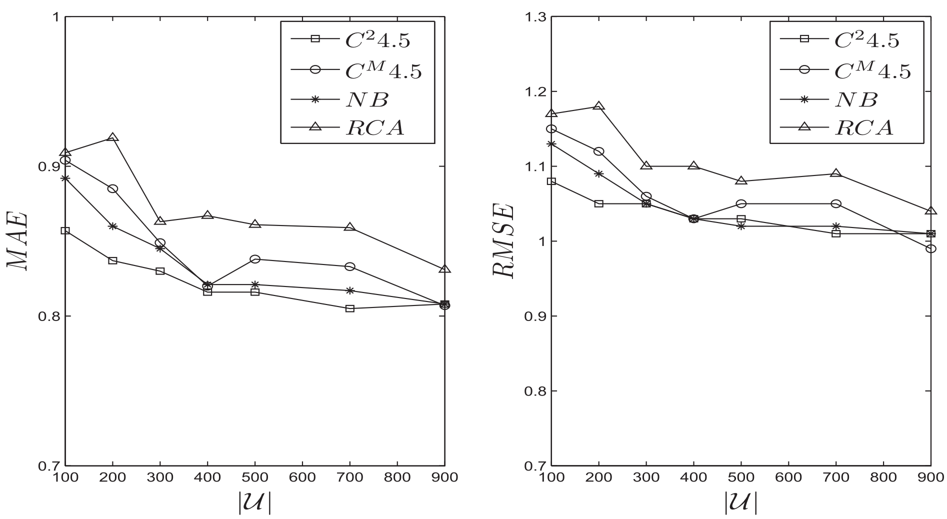

Fig. 5. Results for Scenario 4.

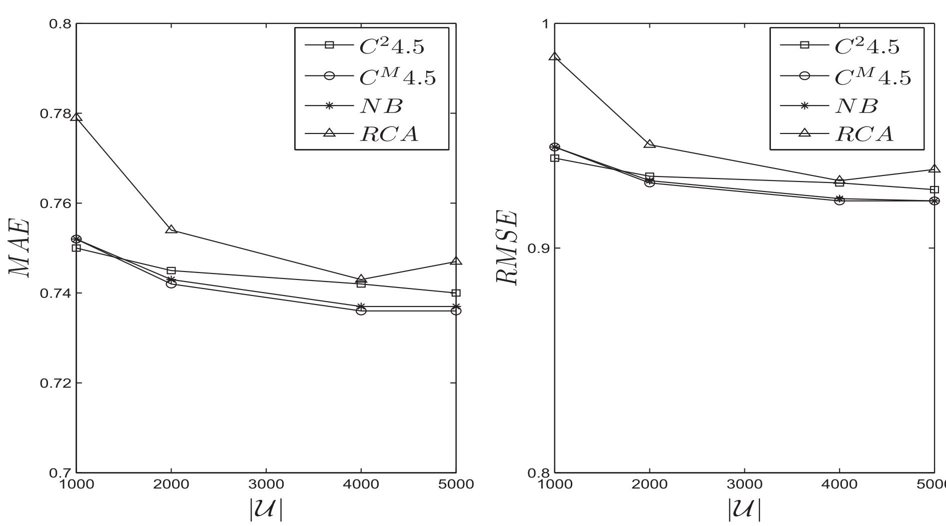

Fig. 6. Results for Scenario 1 – large number of users.

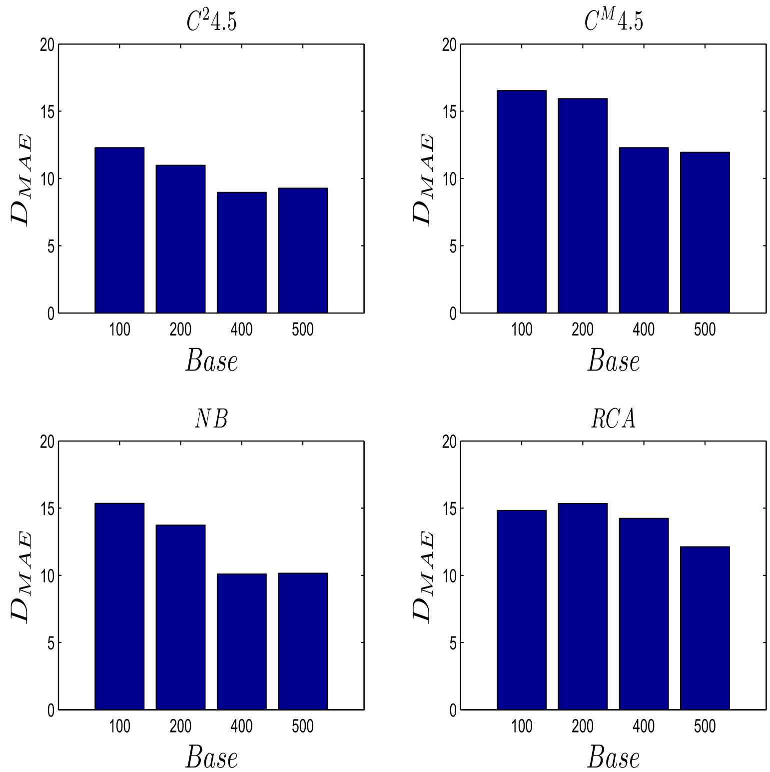

Fig. 7. D_{MAE} comparison.

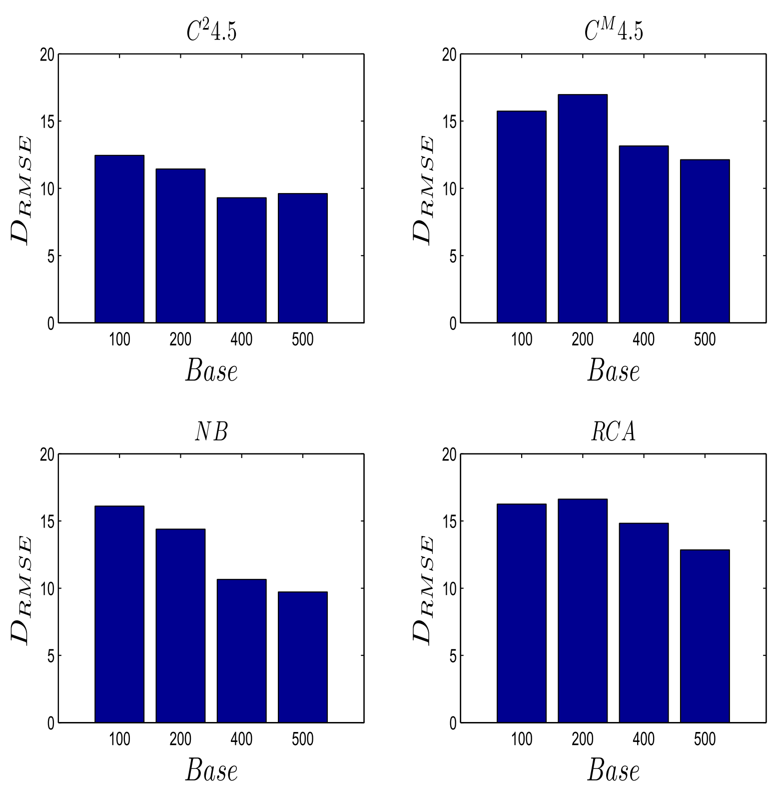

Fig. 8. D_{RMSE} comparison.

recommendations. Scenario 4 is more "fair" as all the demographic data are equally considered.

In Fig. 2, we see our results for the first scenario. We depict both MAE and RMSE. For both metrics, the $C^2$4.5 algorithm exhibits the best performance. As $|\mathcal{U}|$ (number of users) grows, MAE and RMSE are reduced. For $|\mathcal{U}| = 900$, we take MAE approximately equal to 0.8 and RMSE approximately equal to 1.0. As $|\mathcal{U}|$ increases, the system has more data to achieve good performance in the classification process as well in exploiting users demographic information. Hence, the error in the prediction becomes smaller. As expected the *RCA* algorithm performs worse than the rest.

Looking at Figs. 3–5, we see that for the rest of the examined scenarios, we obtain a similar performance for MAE and RMSE. The $C^2$4.5 also performs better compared to the rest of the algorithms. Based on these results, we conclude that weights for each demographic attribute do not play important role for $|\mathcal{U}| \in \{100, 200, \ldots, 900\}$. Hence, we increase the cardinality $|\mathcal{U}| \in \{1000, 2000, 4000, 5000\}$. Fig. 6 depicts our results. Now, the best performance is achieved by the C^M4.5 algorithm accompanied by the *NB*. The minimum MAE value was equal to 0.736 achieved by C^M4.5 when $|\mathcal{U}| = 5000$. The minimum value of *NB* was equal to 0.737 for the same number of users. In average, the $C^2$4.5 algorithm exhibits 0.5% greater MAE value compared to the rest. Through Fig. 6, we see that similar performance is attained for the RMSE metric.

In this point, we consider D_{MAE} and D_{RMSE} for the *MAE* and *RMSE*, respectively. *D* is defined as follows:

$$D = \frac{D_{Base} - D_{Target}}{D_{Base}} \% \qquad (7)$$

We try to reveal the difference in the performance when we adopt small number of users ($|\mathcal{U}| \in \{100, 200, 400, 500\}$) and when we utilize large number of users ($|\mathcal{U}| \in \{1000, 2000, 4000, 5000\}$). D_{Base} stands for $Base \in \{100, 200, 400, 500\}$ and D_{Target} for $Target = 10 \cdot Base$. Fig. 7 is devoted to the presentation of MAE results while Fig. 8 is devoted to the presentation of RMSE results. We see that all the algorithms are affected by the increase in $|\mathcal{U}|$. The $C^2$4.5 algorithm is less affected compared to the rest. The difference in the performance becomes smaller as $|\mathcal{U}|$ increases. However, the difference remains close to 10% as $Base \rightarrow 500$. Concerning the RMSE metric, we see that the C^M4.5 algorithm is heavily affected by the increase in $|\mathcal{U}|$ as in the MAE case. Smaller $|\mathcal{U}|$ leads to greater

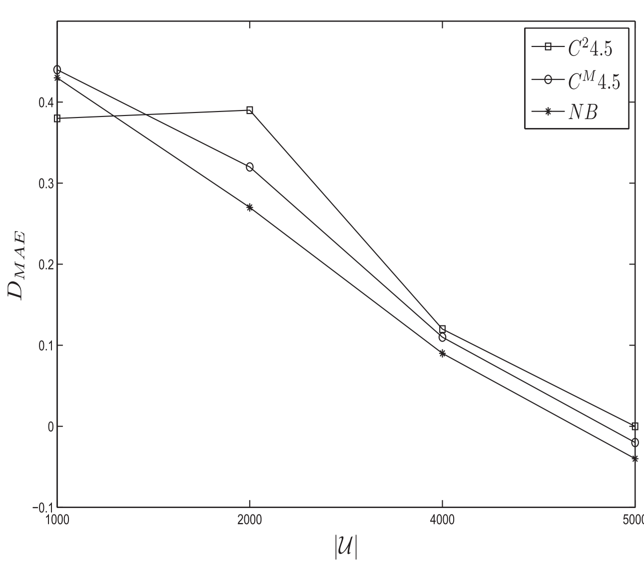

Fig. 9. Results for different ω values – D_{MAE}.

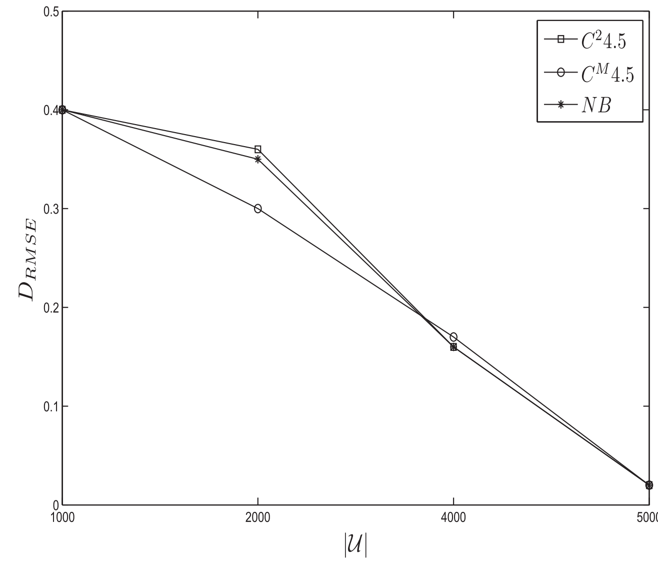

Fig. 10. Results for different ω values – D_{RMSE}.

MAE and RMSE results. This is because the system does not have enough information about users in order to derive better predictions.

Finally, in Figs. 9 and 10, we depict our results for D_{MAE} and D_{RMSE} metrics comparing cases where different ω values are adopted. We remind that the ω parameter is a policy factor affecting the weight of a demographic attribute (taking numeric values). For these results, we take as *Base* the case where $\omega = 15$ and as *Target* the case where $\omega = 0.8$. In the discussed figures, we see that an increased ω value leads to increased MAE values. Better performance is exhibited by the *NB* algorithm as it is less affected by the change in the ω value. When $|\mathcal{U}| = 5000$, we observe that the system performs better when $\omega = 15$. Concerning the RMSE, we see in Fig. 10 that the discussed algorithms exhibit similar behaviour as in the MAE results. Large ω values lead to increased RMSE values.

6. Conclusions

In this paper, we present a method to alleviate the new user cold start problem for RSs applying CF. The proposed system adopts a three-phase approach in order to provide predictions for new users. We adopt a mechanism that takes into consideration their demographic data and based on similarity techniques finds the user's 'neighbours'. We define as 'neighbours', users having similar characteristics with the new user. The idea is that people with a common background and similar characteristics have more possibilities to have similar preferences. Hence, each novel users is classified in a group and accordingly a rating prediction mechanism is responsible to result ratings for items. The final ratings are calculated through a weighted scheme where developers can pay attention to specific attributes or select a more 'fair' approach. Our experimental results show the performance of the proposed techniques. We adopt the dataset provided by the GroupLens research team. The proposed system performs better in cases where a large number of users are already registered in the system. In such cases, the system achieves smaller MAE values increasing the accuracy of ratings prediction.

References

Al Mamunur, R., Istvan, A., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002). Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the seventh international conference on Intelligent user, interfaces.*

Asnicar, F., & Tasso, C. (1997). ifWeb: A prototype of user model-based intelligent agent for documentation filtering and navigation in the word wide web. In *Proceedings of the first international workshop on adaptive systems and user modeling on the world wide web* (pp. 3–12).

Basile, P., de Gemmis, M., Gentile, A., Iaquinta, L., & Lops, P. (2007). The JUMP project: Domain ontologies and linguistic knowledge at work. In *Proceedings of the fourth Italian semantic web applications and perspectives – SWAP 2007*.

Billsus, D., & Pazzani, M. J. (1999). A hybrid user model for news story classification. In *Proceedings of the seventh international conference on user modelling*.

Billsus, D., & Pazzani, M. J. (2000). User modelling for adaptive news access. User modelling and user-adapted interaction, *10* (2–3), pp. 147–180.

Canny, J. (2002). Collaborative filtering with privacy via factor analysis. In *SIGIR*.

Chen, L., & Sycara, K. (1998). WebMate: A personal agent for browsing and searching. In *Proceedings of the second international conference on autonomous agents* (pp. 9–13).

Das, A. et al. (2007). Google news personalization: Scalable online collaborative filtering, WWW (pp. 271–280).

De Gemmis, M., Lops, P., & Semeraro, G. (2007). A content-collaborative recommender that exploitswordnet-based user profiles for neighborhood formation. User modeling and user-adapted interaction. *The Journal of Personalization Research, 17*(3), 217–255.

Deshpande, M. et al. (2004). Item-based top-N recommendation algorithms. *ACM TOIS, 22*(1), 143–177.

Eirinaki, M., Vazirgiannis, M., & Varlamis, I. (2003). SEWeP: Using site semantics and a taxonomy to enhance the web personalization process. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 99–108).

Golbandi, N., Koren, Y., & Lempel, R. (2011). Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth international conference on web search and web data mining*.

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J. (1999). *An algorithmic framework for performing collaborative filtering*. In *SIGIR*.

Jambor, T., & Wang, J. (2010). Optimizing multiple objectives in collaborative filtering. In *Proceedings of RecSys, Barcelona, Spain*.

Jin, R., Si, L., & Zhai, C. (2006). A study of mixture models for collaborative filtering. *Journal of Information Retrieval, 9*(3), 357–382.

Khabbaz, M., & Lakshmanan, L. V. S. (2011). TopRecs: Top-k algorithms for item-based collaborative filtering. In *Proceedings of EDBT, Sweden*.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica, 31*, 249–268.

Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *Proceedings of the second international conference on ubiquitous information management and, communication*.

Lewis, D. D., & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization. In *Proceedings of the third annual symposium on document analysis and, information retrieval* (pp. 81–93).

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, & P. Kantor (Eds.), *Recommender systems handbook* (pp. 73–105). Springer.

Magnini, B., & Strapparava, C. (2001). Improving user modelling with content-based techniques. In *Proceedings of the eighth international conference of user modelling* (pp. 74–83).

Massa, P., & Bhattacharjee, B. (2004). Using trust in recommender systems: An experimental analysis. Trust management, lecture notes in computer science (p. 2995).

Middleton, S. E., Shadbolt, N. R., & De Roure, D. C. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems, 22*(1), 54–88.

Milgram, J., Cheriet, M., & Sabourin, M. (2006). One against one or one against all: Which one is better for handwriting recognition with SVMs?. In *10th International workshop on frontiers in handwriting recognition*.

Mladenic, D. (1999). Machine learning used by personal webwatcher. In *Proceeding of ACAI-99 workshop on machine learning and intelligent agents*.

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on digital libraries* (pp. 195–204).

Moukas, A. (1997). Amalthaea information discovery and filtering using a multiagent evolving ecosystem. *Applied Artificial Intelligence, 11*(5), 437–457.

Oku, K., Hattori, F. (2011). Fusion-based recommender system for improving serendipity. In *Proceedings of the workshop on novelty and diversity in recommender systems, USA*.

Park, S. T., & Chu, W. (2009). Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on recommender systems*.

Pazzani, M. J. & Billsus, D. (2007). Content-based recommendation systems. In P. Brusilovsky, A. Kobsa & W. Nejdl (Eds.), *The adaptive web* (pp. 325–341).

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning, 27*(3), 313–331.

Popescul, A., Ungar, L. H., Pennock, D. M., Lawrence, S. (2001). Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th conference on uncertainty in artificial intelligence*.

Rocchio, J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART system: Experiments in automatic document processing*. Prentice Hall.

Sarvar, B. et al. (2001). Item-based collaborative filtering recommendation algorithms, WWW (pp. 285–295).

Schafer, J., Frankowski, D., Herlocker, J., & Sen, S. (2007) Collaborative filtering recommender systems. In: P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The adaptive web (pp. 291–324).

Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th international ACM SIGIR*.

Shaw, G., Xu, Y., & Geva, S. (2010). Using association rules to solve the cold-start problem in recommender systems. In *Lecture notes in computer science*.

Tso-Sutter, K. H. L., Marinho, L. B. & Schmidt-Thieme, L. (2008). Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of SAC, Brazil*.

Wang, J., de Vries, A. P. & Reinders, M. J. T. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*.

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. In *32nd Annual meeting of the association for, computational linguistics* (pp. 133–138).

Zhang, H. (2004). The optimality of Naive Bayes. In *FLAIRS*.

Zhang, K. & Li, H. (2010). Fusion-based recommender system. In *Proceedings of the 13th conference on information fusion (FUSION)*.

Zhang, Z. K., Liu, C., Zhang, Y., & Zhou, Z. (2010). Solving the cold-start problem in recommender systems with social tags, EPL (Vol. 92).

Zhou, K., Yang, S. H., & Zha, H. (2011). Functional matrix factorizations for cold-start recommendation. In Proceedings of the 34th international ACM SIGIR.