Knowledge-Based Systems 110 (2016) 135–143

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Rating LDA model for collaborative filtering

Xiuze Zhou*, Shunxiang Wu

Department of Automation, Xiamen University, Xiamen City, Fujian, China

ARTICLE INFO

Article history:
Received 22 February 2016
Revised 9 July 2016
Accepted 15 July 2016
Available online 18 July 2016

Keywords:
Interests
RLDA
Collaborative filtering
Users' behavior

ABSTRACT

People are pleased with the great wealth of products in online stores. However, it is more and more difficult for people to choose their favorite products in an online store. Thus, recommendation systems are necessary to provide useful suggestions and selections. A user's choice is not only influenced by his/her interests, but also by the ratings of others. In this paper, we propose a Rating LDA (RLDA) Model for collaborative filtering by adding rating information to the Latent Dirichlet Allocation (LDA). User behavior is not independent; it follows the trend of others. Therefore, we assume that for similar interests, the higher the proportion of high ratings, the more popular the items. We perform experiments on two real world data sets: MovieLens100k and MovieLens1M. Results show that, in terms of F1 score, our proposed approach significantly outperforms some baseline methods.

1. Introduction

With the rapid growth of Internet information, individual processing capacity has become overwhelmed. Thus, recommender systems are necessary to help alleviate the problem of information overload. On one hand, recommender systems help consumers find their favorite products; on the other hand, they help retailers increase sales. These systems have found success in many e-commerce applications, such as Taobao.com and Amazon.com.

Collaborative Filter (CF) and content-based filtering are two strategies widely used in recommender systems for recommending items to users. CF has a wider application because it makes predictions by using only user-item interaction information without additional information or domain knowledge. The key idea of the CF is that users who have had similar preferences in the past are likely to have similar preferences in the future [16,34].

Model-based approaches, such as Singular Value Decomposition (SVD) [35], matrix factorization [19], Bayesian Networks [6], and latent factor models [5,15], use statistical models to learn user interests from user-item rating information and make predictions from the trained model [26].

In all the CF algorithms, matrix factorization methodology [19] is undoubtedly the most popular. Recommendation systems, by using the matrix factorization method, extract a set of latent factors from the comment mode and describe the user and item through these factors' vectors. When there is a match between the current user and some items in these factors, the recommendation systems will recommend these items to this user.

Memory-based algorithms show good accuracy performance, but they cannot handle scalability and the sparsity of data problems. To solve the data sparsity problems, many model-based CF methods have been proposed [31,38]. Model-based CF techniques aim at building a model to represent user rating data and use that model to predict user preference for a specific item. For example, Singular Value Decompositon (SVD) [3] obtains the main factors to reduce dimensionality. Hofmann converts the Latent Semantic model from information retrieval to Collaborative Filtering [15]. These models not only reduce the dimensions of the user-item matrix and smooth out the noise information, but also help the algorithm alleviate data scalability [41].

Recently, Latent Dirichlet Allocation (LDA) has been developed for recommendation systems. Some scholars apply LDA to review texts to obtain additional information for recommendation systems. For example, McAuley and Leskovec [29] first mined user interests from product reviews by using LDA and then combined matrix factorization to predict the unknown item ratings [29]. Wilson et al. [40] utilized LDA to infer the latent properties of items from their textual descriptions and then calculated users' preferences or persona in the same latent topic space based on historical ratings. However, these methods rely on additional text information and can not be applied to the situation without text data. Others used LDA for collaborative filtering directly. For example, Liu et al. [25] achieved the proposed enhancing of collaborative filtering by using the LDA to mine user interests [25]. Zhao et al. [43] used LDA to learn the probability that a user rates an item [43]. These methods can be easily applied to collaborative filtering.

* Corresponding author.
E-mail addresses: xiuzezhou@gmail.com, zhouxiuze@foxmail.com (X. Zhou), wsx1009@163.com (S. Wu).

http://dx.doi.org/10.1016/j.knosys.2016.07.020
0950-7051/

The above approaches capture the hidden connection between users and items. Because there is a latent correlation between some items, there is a certain probability the items will appear together. The most famous example involves diapers and beer. Although there is no direct link between diapers and beer, a grocery store observed that people who buy diapers also buy beer, but there is no direct link between diapers and beer. We use probabilistic models to capture the potential association between commodities.

The main disadvantage of the LDA model is it does not consider the impact that rating information has on recommendation systems. These models consider only whether or not a user buys the item; they neglect an important factor-rating information. Thus, these models cannot infer distribution over their ratings. To deal with this problem, after using the LDA model, Liu et al. [25] proposed iExpand to predict rating scores by using the Pearson Correlation [25]. However, the accuracy of ratings seriously depends on the number of neighbors. Likewise, Zhao et al. [43] proposed a Hybrid approach of Topic Model and Matrix Factorization (HTMMF) by using SVD++ [18] to predict rating scores [43]. This approach improves LDA by using rating behavior to simulate user generated ratings in its first step. However, this step does not consider rating information. Instead, the rating prediction depends entirely on the second step of SVD++.

Traditional approaches work well on the user-item rating data for the prediction of item rating. However, accurate rating prediction does not always result in good recommendation effects [8,24,42]. Thus, many scholars now regard the recommendation problem as a ranking problem [20,24,39]. They model user's interests only on set of items rather than predict the ranking of the items' ratings.

However, rating information is an indispensable factor in recommender systems; recommendation systems rely on users' feedback, especially product reviews and ratings [29]. Item rating information is helpful to improve model performance. The rating trend from previous users strongly infulences current users. For example, when we want to see a film, we will first choose our favorite movie subject. Then, we choose our favorite movies within this subject. But, when we are interested in more than one movie, we tend to choose the film rated higher by other people.

In this paper, we propose a probabilistic latent interest model. To consider the impact of ratings when a consumer selects an item, we incorporate rating information into the LDA model. A user's choice is not only influenced by his/her interests, but also by the ratings of others. The key idea of the RLDA model assumes that, under similar interest, the higher the proportion of high ratings, the more popular the items.

Compared with previous works, the main difference of our model is our model views the ratings and items not separately, but as a whole, resulting in not only retaining rating information, but also reducing the calculation steps. In particular, the proposed algorithm can not only predict the unknown item ratings, but also get user's interest distribution. It is worth noting that our model views the recommendation problem as a ranking problem.

The rest of the paper is organized as follows: Section 2 briefly reviews the background and some related work. Section 3 presents our proposed model in detail. Section 4 introduces the parameter estimation of our model. Section 5 describes experimental settings and results on Movielens data sets to show the performance of our model. Section 6 gives the conclusion and future work.

2. Related work

Before presenting our RLDA model, we briefly review the background and some related work about Collaborative Filtering and the LDA model.

Recommender systems have been widely used in many applications for recommending items to users, such as e-commerce [14,22,36], online news recommendations [17,23], and online movie recommendations [12,21]. Recommender systems have also been successfully adopted for e-business and e-government applications [27,37].

Generally, recommender systems are divided into two basic types: Collaborative Filtering (CF) and Content-based recommendation. The former finds similar users in some groups that have a common interest and then recommends items according to similar user preferences. The latter system learns user' interests from the relevant characteristics of the items that the user commented on in the past and then recommends to the user the items that are similar to his previous favorite items.

Collaborative Filtering (CF). CF uses information about similar user behavior to make recommendations. Besides avoiding the need for collecting extensive information about the items and users, CF requires no domain knowledge and can be applied easily across different recommender systems [7].

CF algorithms compute the similarity among users or items by using a user-item rating matrix. First, many CF approaches compute the similarity among users by comparing rating vectors using Cosine Similarity and Pearson Correlation [13]. Then, the CF algorithms generate predictions by computing a weighted average of the votes of similar users (or similar items) [30,34].

Model based algorithms use the collection of training data to learn a model first and then use the model to generate predictions instead of directly manipulating the original database [9,10,32,33,41].

Latent Semantic models including probabilistic Latent Sematic Analysis (pLSA) and LDA, are generative probabilistic models from collaborative filtering. SVD approaches, as a Singular factor, the highly relevant items that appear together and breaks down the vector into small order approximation matrixes. The key idea of the latent factor model assumes that the similarity among users and items is discovered by lower-dimensional data.

2.1. Latent Dirichlet Allocation

The LDA model is a probabilistic generative model that places objects with similar characteristics into one group. The LDA model has been widely applied to many fields, such as, text topic analysis computer vision, gene sequence recognition and social networks [4]. In recent years, LDA has also been applied to recommendation systems [25,29,40,43].

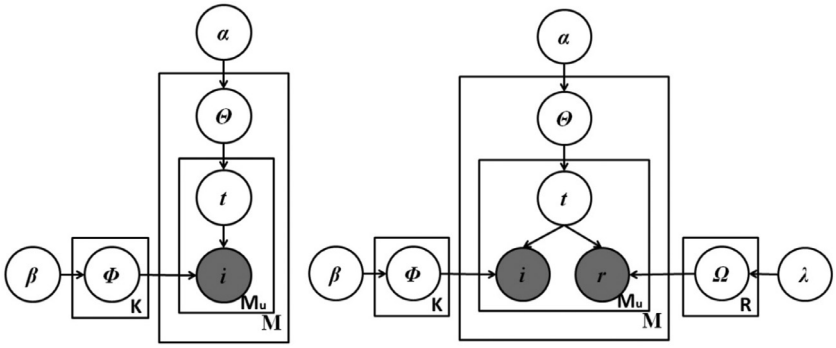
LDA's probabilistic graphical model is represented in Fig. 1(a). From the perspective of collaborative filtering, documents, latent topics, and words are viewed as users, hidden interests, and items, respectively [25]. Items usually occur simultaneously under user hidden interests, just as some words occur simultaneously in documents under latent topics.

In Fig. 1(a), M represents all users. M_u represents all items rated by a user, u, chosen from M. K represents the number of interests. Θ and Φ are the Dirichlet distributions of the models, and α and β are their corresponding two hyper-parameters, respectively.

When choosing an item, i, a user, u, first chooses what he/she is interested in from a multinomial distribution over Θ_u. Then he/she chooses an item, i, from a multinomial distribution over items Φ_t and, using the collapsed Gibbs sampling algorithms[1] to estimate these posterior distributions, obtains the distribution of Θ and Φ:

$$\Theta_{u,t} = P(t|u) = \frac{C_{u,t}^{M,K} + \alpha}{\sum_{t=1}^{K} C_{u,t}^{M,K} + K \cdot \alpha}$$

[1] http://gibbslda.sourforge.net



(a) LDA model (b) RLDA model

Fig. 1. Two probabilistic graphical models. (a) LDA model and (b) RLDA model. In those figures, shaded and unshaded variables denote observed and unobserved variables, respectively. The arrow denotes the conditions of dependency between two variables, and the plate denotes the number of samplings.

$$\Phi_{t,i} = \mathrm{P}(i|t) = \frac{C_{t,i}^{K,N} + \beta}{\sum_{i=1}^{N} C_{t,i}^{K,N} + N \cdot \beta}$$

where M, N and K represent the number of users, items and latent interests, respectively. $C_{u,t}^{M,K}$ represents the number of interests, *t*, assigned to user, *u*. $C_{t,i}^{N,K}$ represents the number of items, *i*, assigned to interest, *t*.

Finally, we obtain the probability of user, *u*, choosing an item, *i*:

$$\mathrm{P}(u,i) = \sum_{k=1}^{K} P(I_i|t=k) \cdot P(t=k|U_u) = \sum_{k=1}^{K} \Phi_{t,i} \cdot \Theta_{u,t}$$

Algorithm 1 The generative process of RLDA modeling.

```
Input: U, I, K, R
  for each interest t ∈ T do
    draw a distribution over items:
      Φ_t ~ Dir(β)
    draw a distribution over ratings:
      Ω_t ~ Dir(λ)
  for each user u ∈ U do
    for each intem i ∈ I do
      assign an interest t given the u:
        t ~ Mul(Θ_u)
      draw an item from the chosen interest:
        i_{u,t}|t ~ Mul(Φ_t)
      draw a rating from the chosen interest:
        r_{u,t}|t ~ Mul(Ω_t)
```

3. The Rating LDA model

In this section, we present our model in detail.

Ratings are a very important factor for enabling recommendation systems to recommend products to consumers. Rating distribution on a particular item will affect a consumer's decision whether or not to buy an item. Benjamin Marlin proposed a User Rating Profile (URP) model, which incorporates semantics at the user level to obtain the user rating profile distribution [28]. Liu et al. introduced LDA to improve CF algorithms [26]. The three layers of LDA represent user, interest, and item in user-item rating. Then Liu et al. used the neighborhood-based method, which computes the Pearson correlation of expanded interests in a vector to predict the rating of items. Zhao et al. used the SVD++ to compute an item's rating [43].

We assume that a user's decision to buy an item not only depends on whether he/she is interested in, but also depends on how many users like it. The ratings of users are not expected to be independent. The rating trend from previous strongly infulences current users. When considering items, if a user is interested in more than one, he/she, following the trends of similar rating behavior, would most likely choose the item that most people like. Thus, the ratio and distribution of each partial score have a significant impact on which item the user will choose.

RLDA extends the previously proposed LDA for collaborative filtering by adding rating information to LDA. A diagram of the RLDA model is given in Fig. 1(b). In the diagram, K, R and M represent the number of interests, rating levels and users respectively. Item and rating information are observed from the user-item matrix. Θ and Φ represent a multinomial distribution over ratings under specific-interest. Ω represents a multinomial distribution over items under specific-interest. α, β and λ represent the symmetric Dirichlet prior of Θ, Φ and Ω respectively.

The key idea of the RLDA model assumes that, in a similar interest, the higher the proportion of high ratings, the more popular the item.

The generative process of the RLDA model is defined in Algorithm 1. First, the model samples the interest, rating-interest, and user-interest distributions according to three Dirichlet hyperparameters. Then, for each item rated by the user, the user first chooses an interest, *t*. Based on *t*, item, *i*, and rating, *r*, are generated independently.

The main difference between the RLDA and the previous approach (iExpand and HTMMF) is that we add the rating information directly to our model, rather than calculate it separately.

4. Parameter estimation

To evaluate our proposed approach, we perform experiments on Movielens data sets. Then, we present experimental results and analyze the effectiveness of those approaches in comparison with other baselines, LDA, iExpand, SVD_Pure and SVD_Neib.

The three parameters, user interests (Θ), interest items (Φ) and interest rating (Ω) are intractable latent variables for obtaining the exact parameters. Therefore, we choose the collapsed Gibbs sampling algorithm to estimate these posterior distributions. Gibbs sampling, a simple and effective algorithm to estimate parameters in the LDA family, is a special form of the Markov chain Mote Carlo (MCMC) [11].

Based on the independent assumptions in the RLDA model, the joint distribution of interests, items and ratings is defined as follows:

$$\begin{aligned}\mathrm{P}(t,i,r|\alpha,\beta,\lambda) &= P(t|\alpha,\beta,\lambda) \cdot P(i,r|\alpha,\beta,\lambda) \\ &= P(t) \cdot P(i|t) \cdot P(r|t) = \Theta_t \cdot \Phi_{t,i} \cdot \Omega_{t,r}\end{aligned}$$

In in RLDA model, α, β and λ are Dirichlet priors. Θ, Φ, and Ω represent multinomial distributions sampled from the Dirichlet distribution. We must estimate the models' three latent variables and derive their conditional probability for the current state j, i.e.

$$P\left(T_j^k, I_j^i, R_j^r \middle| U_j^u, \alpha, \beta, \lambda\right) \propto \frac{n_{u,-j}^t + \alpha}{\sum_{t=1}^{T} n_{u,-j}^t + T \cdot \alpha} \cdot \frac{n_{t,-j}^i + \beta}{\sum_{i=1}^{N} n_{t,-j}^i + N \cdot \beta} \cdot \frac{n_{t,-j}^r + \lambda}{\sum_{r=1}^{R} n_{t,-j}^r + R \cdot \lambda}$$

where, the notion T_j^k, I_j^i, R_j^r and U_j^u represent the kth interest, the ith item, the rth rating, and the uth user, respectively. T, N, and R represent the number of interests, items, and ratings, respectively. The notations $n_{u,-j}^t, n_{t,-j}^i$ and $n_{t,-j}^r$ represent the number of interests assigned to user, u, the number of items assigned to interest, t, and the number of ratings assigned to interest, t, respectively. Finally, we obtain the parameters of Θ, Φ, and Ω:

$$\Theta_{u,t} = \frac{n_u^t + \alpha}{\sum_{t=1}^{T} n_u^t + T \cdot \alpha}$$

$$\Phi_{u,t} = \frac{n_t^i + \beta}{\sum_{i=1}^{N} n_t^i + N \cdot \beta}$$

$$\Omega_{u,t} = \frac{n_t^r + \lambda}{\sum_{r=1}^{R} n_t^r + R \cdot \lambda}$$

After we determining the model parameters, we infer P(i,r|u) and then rank the items for a given user according to P(i,r|u).

5. Experiments

In this section, we describe the experimental settings in detail.

5.1. Data set

To present our model's performance, we performed experiments on two public data sets: MovieLens100K and MovieLens1M. These two data sets are collected, and the movie rating data sets are made available from the MovieLens web site[2]. The MovieLens100K data set consists of 100,000 ratings from 943 users and 1682 movies. On average, each user watched 106 movies. The MovieLens1M data set consists of 1000,000 ratings from 6040 users and 3952 movies. On average, each user watched 166 movies. The data sets are randomly divided into 80% training data and 20% testing data.

5.2. Baseline approaches

We compare the proposed RLDA model with four mode-based approaches as follows:

- **LDA:** View the item as the word, and review the user as the document to ascertain the P(I|U) for recommendation.
- **iExpand:** The first step is based on the LDA model, and the second step uses the Cosine Similarity on expanded user interest to predict the unknown rating [25].
- **SVD_Pure:** The training user-item matrix is decomposed into three component matrixes with f features: $R_f = U_f \cdot S_f \cdot V_f^T$. Then, three component matrixes, U_f, S_f and V_f^T are used directly to predict the unknown rating.
- **SVD_Neib:** First, we use the SVD approsach on the training user-item matrix. Then, we choose the Cosine Similarity on users (KNN = 20) to predict the unknown rating. This method is now the state-of-the art method for regular CF tasks, and was used by the winner of the Netflix prize [1,2].

[2] http://movielens.org

5.3. Parameter setting

For all expoerimtents with the LDA family models, we fixed the hyper-parameters, α and β, at 50/T and 0.001, respectively, and set λ at 0.01 for RLDA. We ran the Gibbs sampling for 1000 iterations. The number of interests is set from 10 to 300 on the MovieLens data sets.

For the SVD family, we set the number of nearest neighbor between users at twenty (KNN = 20–50 is an ideal parameter setting according to Koren's research), and set the latent factor space of dimensionality, f, from 10 to 300.

For all experiments, we randomly ran ten times. Finally, we obtain the TOP-N ($N = 10$) list for recommendations.

5.4. Performance evaluation metrics

The Mean Absolute Error (MAE) and the Root Mean Error (RMSE) characterize prediction accuracy. However, a low prediction error does not mean a high recommendation quality. For example, for a test score of 4, prediction scores of 5 and 3 have the same prediction error, but their position on the recommendation list is different from that of a Five-point scale system. Thus, the TOP-N recommendation performance is more meaningful for recommendation systems. We choose the F1 score to evaluate the performance of all algorithms.

- **Precision:** the ratio of the number of relevant items (hits) on the TOPN list to N.
- **Recall:** the ratio of the number of relevant items (hits) on the TOPN list to the number of test data of u.
- **F1 score:** harmonic mean of precision and recall

$$\text{precision} = \frac{hits}{N}$$

$$\text{recall} = \frac{hits}{|Test_u|}$$

$$\text{F1} = \frac{2 \times precision \times recall}{precision + recall}$$

where N is the number of TOP-N items on the recommended list. $|Test_u|$ is the number of test data from u. The higher the F1 score, the better.

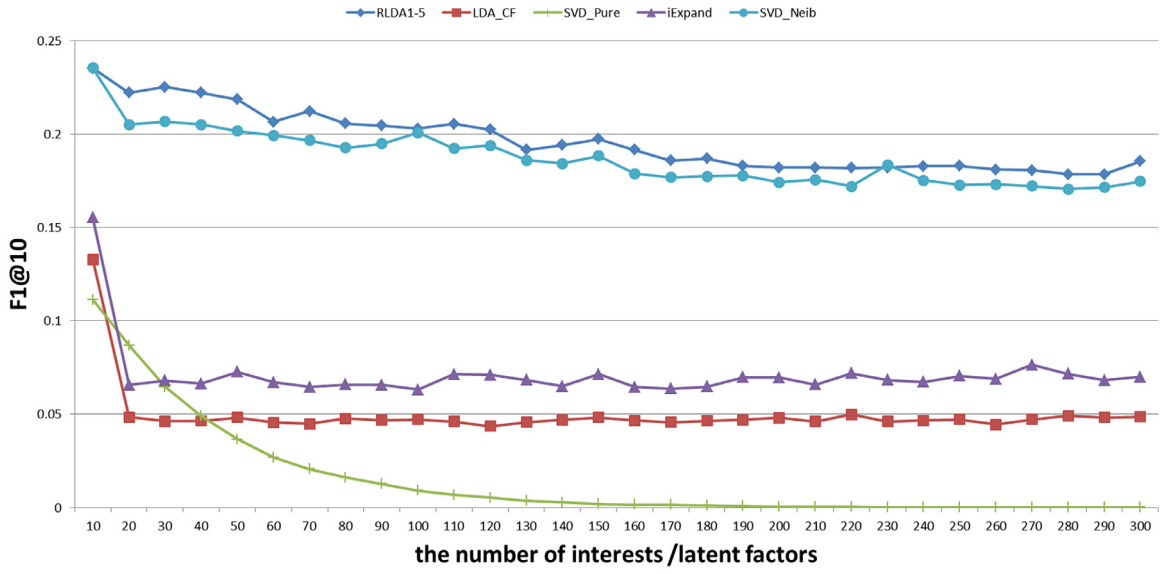
5.5. Results and analysis

To empirically evaluate RLDA quantitatively, we performed experiments on real world data sets and compared our model with baselines.

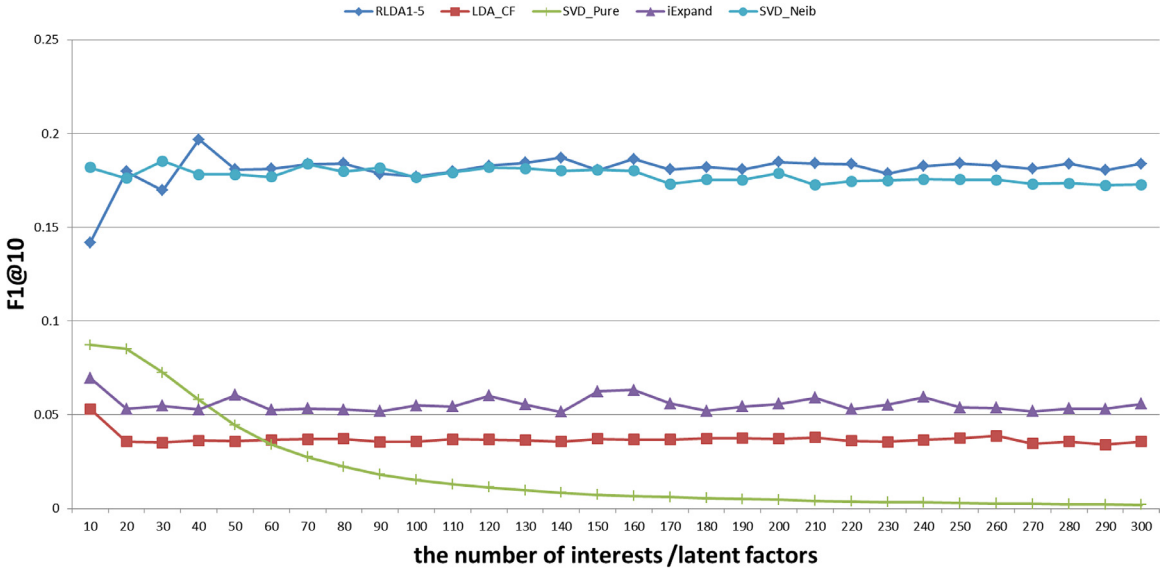
For the number of interests or latent factors ranging from 10 to 300, the F1@10 of these approaches on the MovieLens100K and MovieLens1M data sets is shown in Fig. 2.

RLDA5 indicates that RLDA recommends the items, ranked by their rating probability, that have a rating of five. Similarly, RLDAm-n indicates that RLDA recommends the items, ranked by their rating probability, that have a rating from m to n.

From the experimental results, it is seen that the number of interests or latent factors affects the performance of every approach, especially the SVD_Pure approach. On one hand, because too small a value of k causes these approaches to be under-fitted,when the value of k is very small (less than 60), these approaches all perform well as the value of k increases. On the other hand, because a large value of k causes these approaches to be over-fitted, when the value of k is large, as the value of k increases, the performance for all of these approaches worsens. Consequently, to achieve optimum recommendation performance, it is essential to choose an appropriate value of k for the approaches.



(a) Performance on MovieLens100K



(b) Performance on MovieLens1M

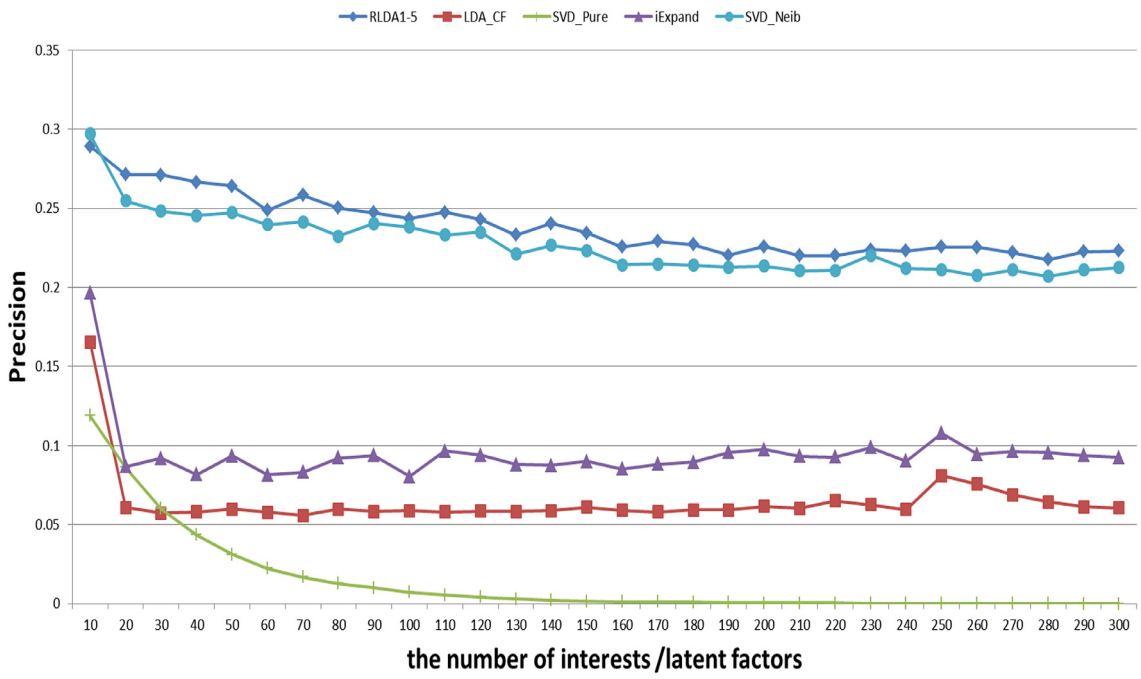
Fig. 2. The changes of F1@10 for MovieLens100K and MovieLens1M when k increases from 10 to 300. We can find the proposed RLDA model outperforms the other approaches under different k. Moreover, RLDA model has a much better performance than other approaches especially when k is large.

From Fig. 2, the results indicate that, in most cases, our proposed method performs best. As the number of interests/latent factors increases, RLDA mimics SVD_Neib, and both perform significantly better compared with the other approaches, especially when the number of interests/latent factors is large. But, the performance of SVD_Neib is differnent with a different number of neighboods. An increase in the number of interest/latent factors has little effect on the F1@10 results of RLDA and SVD_Neib, but others drop sharply. A large number of interests/latent factors leads to the over-fitting of SVD_Pure; however, SVD_Neib is very robust because it strengthens its generalization ability by adding a neighborhood to SVD. The performances of LDA and iExpand mimic each other because iExpand, based on the LDA model, is limited by the LDA performance.

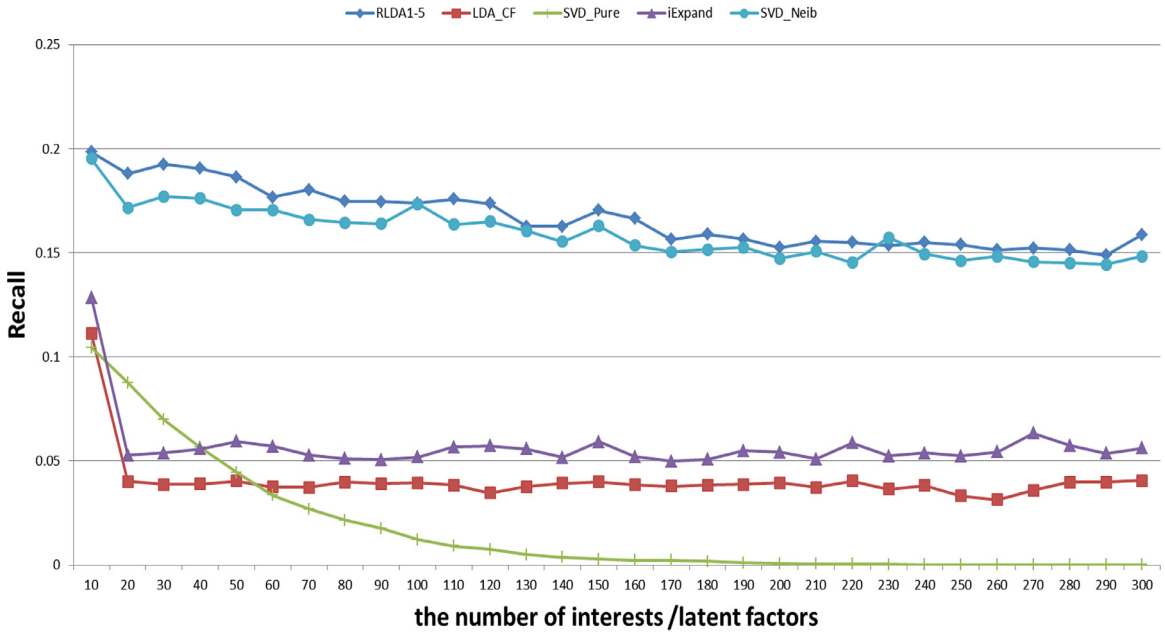
From Fig. 2, we also see that the performance of MovieLens100k is better than that of MovieLens1M. The reason is the data of the latter is more sparse than that of the former. The rating matrix densities of MovieLens100K and MovieLens1M are 6.3% and 4.2%, respectively. All of these models suffer from a data sparsity problem. As a result, the experimental results indicate that our model is more stable than the baseline approaches.

For the number of interests or latent factors (k) ranging from 10 to 300, the Precision and Recall of these approaches on the MovieLens100K data sets is shown in Fig. 3. We find that the proposed RLDA model outperforms the other approaches under different k. Moreover, the Precision, Recall and F1@10 of these approaches mimic each other.

Fig. 4 shows the performance of the RLDA family for the MovieLens100k and MovieLens1M. Regarding the F1 score, from Fig. 4 we see that F1(RLDA5) < F1(RLDA4-5) < F1(RLDA3-5) < F1(RLDA2-5) < F1 (RLDA1-5). These curves are similar to each



(a) The Precision performance on Movielens100K



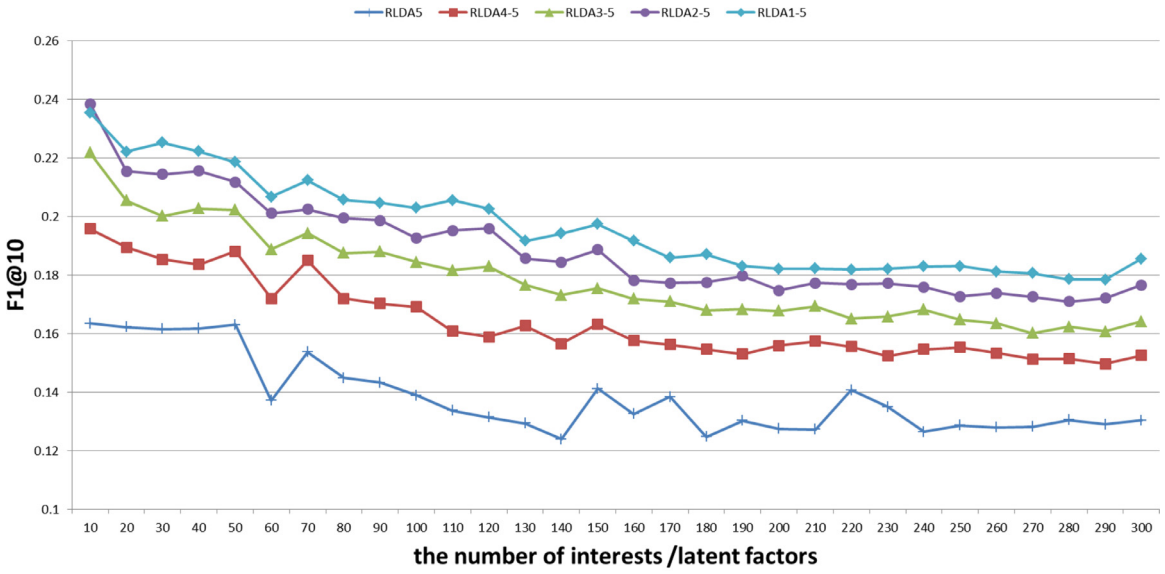
(b) The Recall performance on Movielens100K

Fig. 3. The changes of Precision and Recall for MovieLens100K when k increases from 10 to 300.

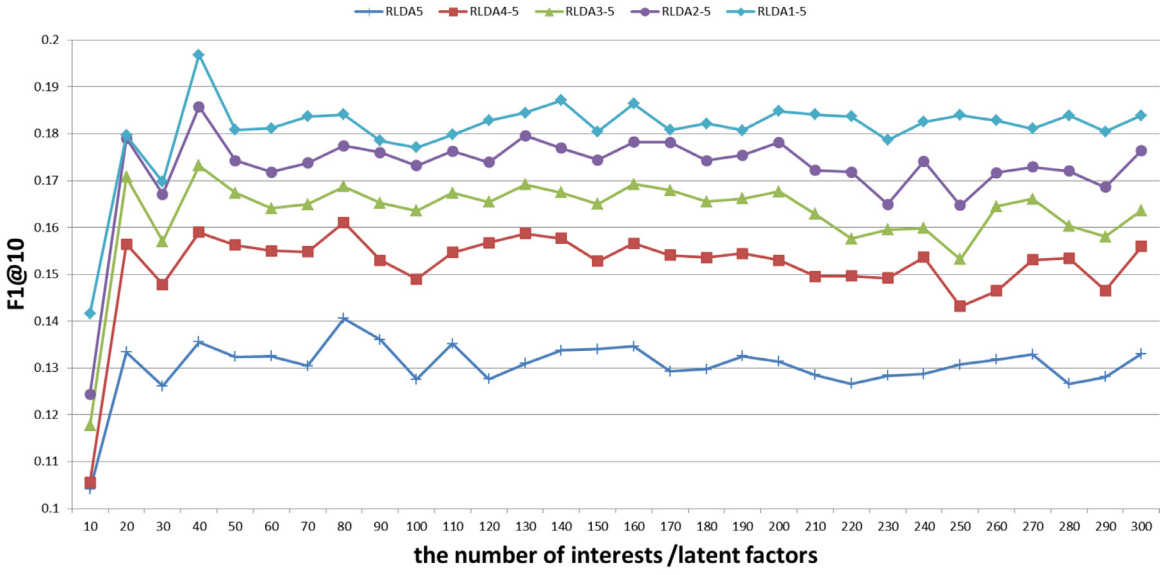
other. Furthermore, from the bottom up, the gap between any two curves decreases. Thus, we conclude the following:

1. Ratings do affect the recommended accuracy;
2. The approach that considers rating information achieves a higher accuracy than that without considering rating information;
3. The lower the score, the fewer the number of items recommended, but the recommendation accuracy still improves;
4. Recommending the highest rating does not result in the best recommended accuracy;
5. Users are interested in some items with a lower rating if these items fall in the users areas of interest;

Then, we futher compared our approach with other baseline approaches with different aparsity level under different latent factor/interests. We choose two cases, $k = 10$ and $k = 100$, representing the small and large value of the latent factor/interests. A comparision between our method and baselines is given in Fig. 5. For example, the training set ratio (%) 20 denotes that we randomly choose 20% of all user-item rating data for training and used remaining 80% for evaluation.



(a) Performance of RLDA family on MovieLens100K



(b) Performance of RLDA family on MovieLens1M

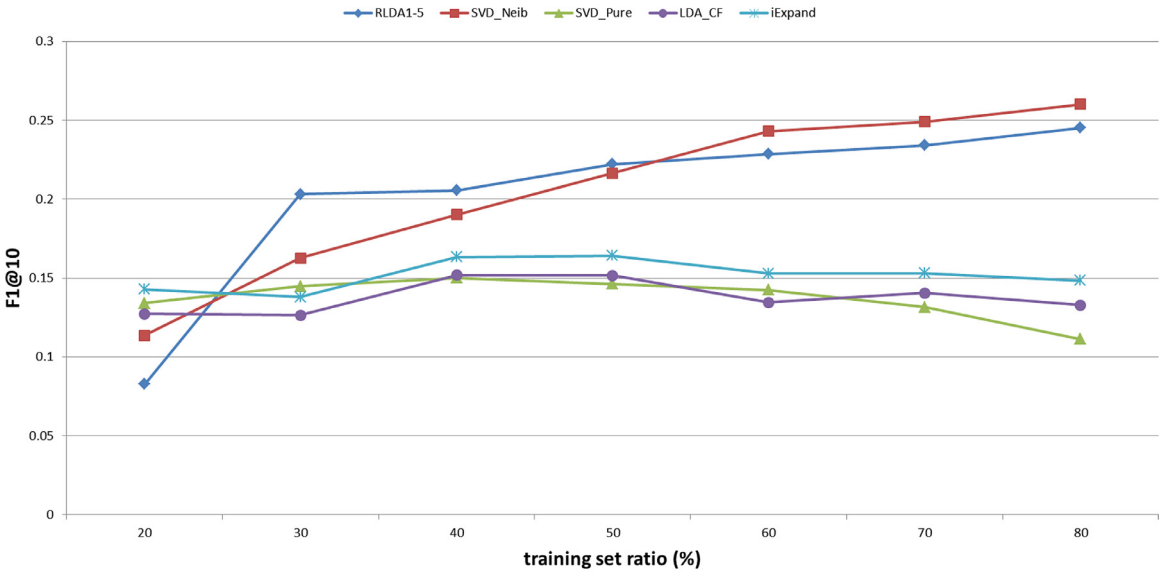
Fig. 4. The changes of F1@10 of RLDA family for MovieLens100K and MovieLens1M when k increases from 10 to 300. We can find the RLDA1-5 model outperforms the other approaches under different k. Recommended items with lowest rating will reduce the recommendation accuracy.

As the latent factor/interests increases, the values of RLDA1-5 and iExpand increase. Other methods do not change significantly. Because the Movielens100k data set it is very sparse. The data sparsity problem seriously affects the performance of these methods, especially the LDA, SVD_Pure and iExpand methods. When the training set ratio (%) is large than 60, the SVD_Neib method performs better than the RLDA1-5 method under the latent factor/interests $k = 10$. Hoever, the RLDA1-5 method performs better than the SVD_Neib method, in most case, when the latent factor/interests $k = 100$. For both approaches, increasing the user-item rating data significantly improves their accuracy. For RLDA1-5, the probability remains constant that a latent link appears more stable. For SVD_Neib, increasing the user-item raing data adds additonal neighbors, resulting in improved prediction accuracy.

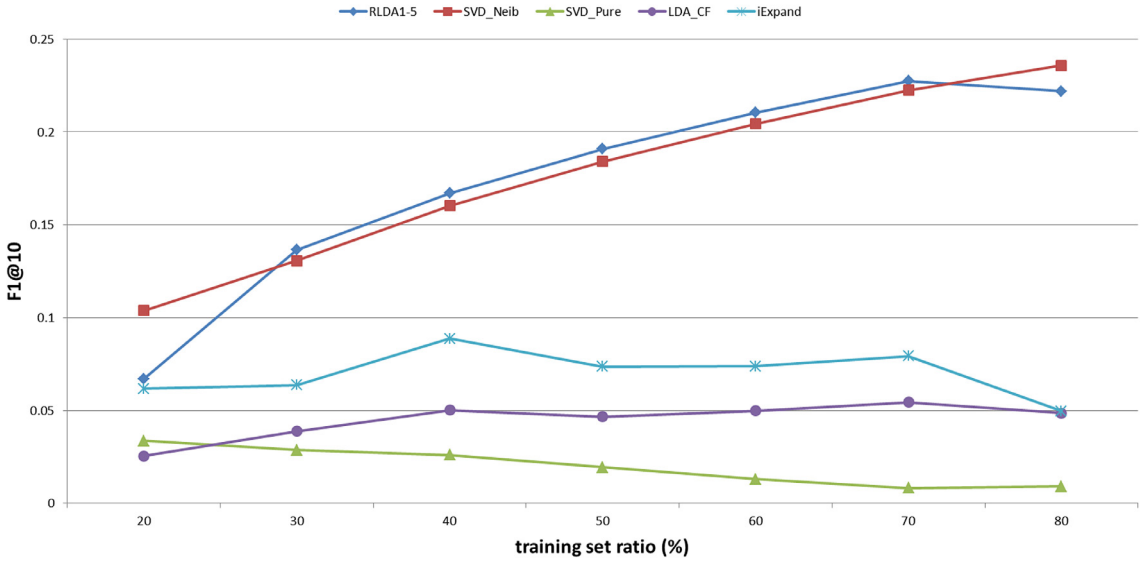
6. Conclusion

This paper proposed a novel model, Rating LDA (RLDA) model, for collaborative filtering, which extends the existing LDA model by adding rating information. The RLDA model for collaborative filtering provides a relatively simple probabilistic model to explore the relationships between users, interests and ratings. Item rating information, crucial for recommendations, helps improve the LDA model's performance. The rating trend from previous users provides strong guidance for current users.

We conducted experiments on two real world data sets, MovieLens100K and MovieLens1M. The results demonstrated that the RLDA model not only achieves better TOPN item recommendations than the other baseline approaches, but also simultaneously obtains the user's rating information. Furthermore, our model is simple, easy to implement, and scales up well. The experimental re-



(a) The latent factor/interests k=10



(b) The latent factor/interests k=100

Fig. 5. The value of F1@10 of all approaches on MovieLens100K data set when the training set ratio (%) x increases from 20 to 80.

sults show that one system cannot obtain the highest accuracy by recommending only the items with the highest scores. Likewise, although the recommendation systems recommend to users items with a minimum score, it is still possible to improve the accuracy. Consequently, choosing a suitable recommendation rating is essential for a recommendation system.

In future work, we will further develop our model by adding some extra information to obtain recommendations with higher accuracy. Because many models now are unable to capture the latest changes in user preferences over time, in future work, we will consider parallel processing for the online recommender systems.

Acknowledgment

The authors would like to thank Michael McAllister for proofreading this paper.

References

[1] R.M. Bell, Y. Koren, C. Volinsky, The BellKor Solution to the Netflix Prize, 2007.
[2] J. Bennett, S. Lanning, The Netflix prize, KDD Cup Workshop Conjunction KDD, 2007.
[3] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Presented at the ICML '98, Morgan Kaufmann Publishers, 1998, pp. 46–54.
[4] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (2012) 77–84.
[5] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[6] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
[7] W.Y. Chen, J.C. Chu, J. Luan, H. Bai, Y. Wang, E.Y. Chang, Collaborative filtering for orkut communities: discovery of user latent behavior, in: Presented at the International World Wide Web Conference, New York, NY, USA ©2009, ACM, 2009, pp. 681–690.
[8] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Presented at the Proceedings of the fourth ACM Conference on Recommender Systems, Barcelona, Spain, 2010, pp. 39–46, doi:10.1145/1864708.1864721.

[9] P. De Meo, F. Messina, D. Rosaci, G.M.L. Sarné, Recommending Users in Social Networks by Integrating Local and Global Reputation, in: Presented at the 7th International Conference on Internet and Distributed Computering Systems, C a labria, Italy, Springer International Publishing, 2014, pp. 437–446, doi:10.1007/978-3-319-11692-1_37.
[10] P. De Meo, A. Nocera, D. Rosaci, D. Ursino, Recommendation of reliable users, social networks and high-quality resources in a social Internet working system, AI Commun. 24 (2011) 31–50, doi:10.3233/AIC-2010-0484.
[11] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, in: Pattern Anal. Mach. Intell. IEEE Trans. On, 1984, pp. 721–741.
[12] F.M. Harper, X. Li, Y. Chen, J.A. Konstan, An economic model of user rating in an online recommender system, in: User Modeling 2005, Berlin , Heidelberg, Springer, 2005, pp. 307–316.
[13] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Presented at the Research and Development in Information Retrieval, New York, NY, USA ©1999, ACM, 1999, pp. 230–237, doi:10.1145/312624.312682.
[14] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. TOIS 22 (2004) 5–53.
[15] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. TOIS 22 (2004) 89–115.
[16] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inf. Retr. 4 (2001) 133–151, doi:10.1023/A:1011419012209.
[17] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, J. Riedl, GroupLens: applying collaborative filtering to usenet news, Commun. ACM 40 (1997) 77–87.
[18] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Presented at the Knowledge Discovery and Data Mining, New York, NY, USA, ACM, 2008, pp. 426–434.
[19] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (2009) 30–37.
[20] Y. Koren, J. Sill, OrdRec: an ordinal model for predicting personalized item rating distributions, in: Presented at the Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 117–124.
[21] F. Lin, X.Z. Zhou, W.H. Zeng, Sparse online learning for collaborative filtering, Int. J. Comput. Commun. Control 11 (2016) 248–258.
[22] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (2003) 76–80, doi:10.1109/MIC.2003.1167344.
[23] J.H. Liu, P. Dolan, E.R. Pedersen, Personalized news recommendation based on click behavior, in: Presented at the Proceedings of the 15th International Conference on Intelligent User Interfaces, New York, NY, USA ©2010, 2010, pp. 31–40, doi:10.1145/1719970.1719976.
[24] N.N. Liu, M. Zhao, Q. Yang, Probabilistic latent preference analysis for collaborative filtering, in: Presented at the Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 759–766, doi:10.1145/1645953.1646050.
[25] Q. Liu, E.H. Chen, H. Xiong, C.H.Q. Ding, J. Chen, Enhancing collaborative filtering by user interest expansion via personalized ranking, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2012) 218–233, doi:10.1109/TSMCB.2011.2163711.
[26] Q. Liu, Y. Xiong, W. Huang, Combining user-based and item-based models for collaborative filtering using stacked regression, Chin. J. Electron. 23 (2014).
[27] J. Lu, Q. Shambour, Y. Xu, Q. Lin, G. Zhang, BizSeeker: a hybrid semantic recommendation system for personalized government-to-business e-services, Internet Res 20 (2010) 342–365, doi:10.1108/10662241011050740.
[28] B.M. Marlin, Modeling user rating profiles for collaborative filtering, Advances in Neural Information Processing Systems, 2003.
[29] J. McAuley, J. Leskovec, in: Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text, ACM Press, 2013, pp. 165–172, doi:10.1145/2507157.2507163.
[30] P. Melville, V. Sindhwani, Recommender systems, Encycl. Mach. Learn. (2010) 829–838, doi:10.1007/978-0-387-30164-8_705.
[31] K. Miyahara, M.J. Pazzani, Improvement of collaborative fltering with the simple bayesian classifer, Inf. Process. Soc. Jpn. 43 (2002) 3429–3437.
[32] D. Rosaci, CILIOS: Connectionist inductive learning and inter-ontology similarities for recommending information agents, Inf. Syst. 32 (2007) 793–825, doi:10.1016/j.is.2006.06.003.
[33] D. Rosaci, G.M.L. Sarné, Recommending multimedia web services in a multi-device environment, Inf. Syst. 38 (2013) 198–212, doi:10.1016/j.is.2012.08.002.
[34] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, in: Item-Based Collaborative Filtering Recommendation Algorithms, ACM Press, 2001, pp. 285–295, doi:10.1145/371920.372071.
[35] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Application of Dimensionality Reduction in Recommender System-A Case Study, 2000 DTIC Document.
[36] J.B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in: Presented at the Proceedings of the 1st ACM conference on Electronic commerce, 1999, pp. 158–166, doi:10.1145/336992.337035.
[37] Q. Shambour, J. Lu, A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services, Int. J. Intell. Syst. 26 (2011) 814–843, doi:10.1002/int.20495.
[38] L.H. Ungar, P.F. Foster, Clustering methods for collaborative filtering, Presented at the Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence, AAAI Press, 1998.
[39] M. Weimer, A. Karatzoglou, Q.V. Le, A. Smola, Cofi rank-maximum margin matrix factorization for collaborative ranking, Adv. Neural Inf. Process. Syst. (2007) 1593–1600.
[40] J. Wilson, S. Chaudhury, B. Lall, P. Kapadia, Improving collaborative filtering based recommenders using topic modelling, in: Presented at the Web nce, Washington, DC, USA, IEEE Computer Society, 2014, pp. 340–346.
[41] B. Xu, J. Bu, C. Chen, An exploration of improving collaborative recommender systems via user-item subgroups, in: Presented at the WWW 2012, 2012, pp. 21–30.
[42] X. Zhao, Z. Niu, W. Chen, Interest before liking: two-step recommendation approaches, Knowl.-Based Syst. 48 (2013) 46–56, doi:10.1016/j.knosys.2013.04.009.
[43] X. Zhao, Z. Niu, W. Chen, C. Shi, K. Niu, D. Liu, A hybrid approach of topic model and matrix factorization based on two-step recommendation framework, J. Intell. Inf. Syst. 44 (2015) 335–353, doi:10.1007/s10844-014-0334-3.