

# Integrating Opinion Leader and User Preference for Recommendation

Dong Wu<sup>1</sup>, Kai Yang<sup>2</sup>, Tao Wang<sup>2</sup>, Weiang Luo<sup>2</sup>,  
Huaqing Min<sup>2</sup>(✉), and Yi Cai<sup>2</sup>

<sup>1</sup> School of Information Science and Technology, Lingnan Normal College,  
Chikan District, Zhanjiang 524048, Guangdong, China

<sup>2</sup> School of Software Engineering, South China University of Technology,  
Panyu District, Guangzhou 510006, Guangdong, China  
hqmin@scut.edu.cn

**Abstract.** Collaborative filtering (CF) is one of the most well-known and commonly used technology for recommender systems. However, it suffers from inherent issues such as data sparsity. Many works have been done by used additional information such as user attributes, tags and social relationships to address these problems. We proposed an algorithm named *OLrs* (Opinion Leaders for Recommender System) based on the trust relationships. Specifically, the opinion leaders who have a strong influence for the active user and an accurate evaluation of the recommend item will be identified. The prediction for a given item is generated by ratings of these opinion leaders and the active user. Experimental results based on Epinions data set demonstrated that the prediction accuracy of our method outperforms other approach.

**Keywords:** Recommender systems · Data sparsity · Opinion leader · Matrix factorization

## 1 Introduction

As the rapid development of information technology and extensive application of information services, people's daily behavior such as news reading, buddy communication, shopping and payments can be made online. It gained unprecedented convenience, but also caught in a serious problem how to find valuable information or find goods to meet the requirements. This is known as information overload problem. In order to effectively solve the problem, the recommender system was designed and has begun to be applied to various fields. Especially, it has played a good role in the large-scale e-commerce shopping guide website [1].

Collaborative filtering (CF) is one of the most well-known and commonly used technology for recommender systems. Many successful cases show that good results can be got using CF. There is a basic assumption that items appreciated by people similar to someone will also be appreciated by that person. As pointed in [1] that if this assumption is not satisfied, there will be some defects and data

sparsity is the most serious one. The data sparsity problem is the problem of having too few ratings, and hence, it is difficult to identifying similar users for the target user or similar items for the target item [2]. It was pointed out in [3] that the data sparsity problem is certainly exist when CF depends only on the user-item rating. Because only a small number of users will give item rating when using the actual recommender systems.

Currently, there are many methods have been proposed and data sparsity problem is resolved with the additional information such as context information [4], tags [5], social relationships [6]. The performance of these methods is better than CF.

Actually, there are opinion leaders in social networks especially trust networks. Some behavior or opinions of these opinion leaders often influence the behavior or comments of others in the network [7]. A choice made by ordinary people in trust networks is in fact a combination of views of opinion leaders and themselves. We can use this influence as the additional information to solve data sparsity problem in the recommender system. Thus, we proposed an algorithm named *OLrs* (Opinion Leaders for Recommender System) in this paper.

To determine whether an item such as a book is recommended to *Alan*, we have to predict the rating of this book *Alan* will be given. If the rating is highly, we can make recommend. The rating of the book given by *Alan* consists of two parts: one is based on *Alan's* own experience and another is based on the views of the opinion leaders who has a strong influence for *Alan* and accurate evaluation of the book. In our work, the former was calculated using tag-based recommender algorithm and the latter was calculated by doing the following things. First, the trust network of *Alan* was established based on trust relationships between users in the recommender system. Second, opinion leaders who can make an accurate evaluation on the book were found out from the most influential opinion leaders. Finally, the ratings of these opinion leaders are aggregated. We only used the views of the opinion leaders who have a strong influence for *Alan* and an accurate evaluation of the book. That is the difference between our method and the former algorithms in which the views of all opinion leaders who have influence for *Alan* are used.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related works. Section 3 presents the *OLrs* method and Sect. 4 to verify the performance of present work, and compare with the state-of-the-art methods. We conclude the study in Sect. 5.

## 2 Related Work

Many works have been done in the literature to resolve the data sparsity problem, which is cause by lack of user-item ratings. These proposed methods can be classified into two categories: one is using CF method after filling the missing ratings with the average of existing ratings or the median rating, and another is merge various additional information, such as user attributes [8], multimedia

content [9], and location information [10] into CF to improve recommendation performance. Many works show that the latter is effective than the former. Our proposed method in this paper is follow in the second way as well.

Recently, latent factor model has become popular. As the Netflix Prize competition has demonstrated, Matrix factorization model (MF) [11] are superior to classic CF for producing product recommendations. Salakhutdinov and Mnih [12] propose a new model name Probabilistic Matrix Factorization (PMF) based on Bayesian deduction. Singh and Gordon [13] propose collective matrix factorization (CMF), which simultaneously factorizes the user-item rating matrix and user attributes matrix to address the data sparsity problem better. However, these latent factor models face real difficulties to explain predictions.

Except for the attributes of user or item, there is other information created by user, such as Tags can be used to strengthen the recommendation. Tags are short labels used to mark the item [5, 14]. They are created according to the users' needs. The frequency of using some tags can reflect the user's interest in hobbies. This feature is used by researchers to improve the quality of recommendation. Tags are used to connect users and items in [15–17]. User-item rating matrix and tag-based user-user similarity matrix are used to factorize in the algorithm proposed in [18].

So many social relationships such as friendship, membership and social trust are applied to the CF with the emergence of a large number of social networking sites. And social trust is proved to be better than other social relationships to improve the performance of CF. Trust can be classified into two categories: implicit trust (e.g., [19, 20]) and explicit trust (e.g., [21, 22]). The former is inferred from user behaviors such as ratings whereas the latter is directly specified by users. Researchers considered that explicit trust is more reliable to implicit trust. TidlTrust [23] is a modified breadth-first search algorithm in a trust network. In this algorithm, it was considered highly trusted users should appear in the search of the shortest path. And the user's rating on item could be calculated by the ratings and degree of be trusted of these people. MoleTrust [24] is similar to TidlTrust. But depth-first search is used and the users whose depth smaller than a threshold are trusted in this algorithm. TrustWalker [25] is an algorithm in which random walk method is used to search trusted people in trust network. In all these algorithms there is one thing in common and it is searching trusted people in trust network. These trusted people are actually opinion leaders and their behavior or comments become the reference of other users in the network. But when users rate one item, they will make a conclusion not only according to the comment of opinion leaders, but also according to their own experience. And this is not taken account in the above work. The main idea of our work is getting the users final rating according to the ratings of opinion leaders and their own experience to improve data sparsity problem. These opinion leaders must have a strong influence for the user and an accurate evaluation of item.

### 3 The *OLrs* method

In this section, we propose an effective approach named *OLrs*, in which integrating the opinion leaders and user preference to predict the rating of active user for an item. Two steps are taken to make recommendations. First, the opinion leaders of the active user are identified. Second, recommendations are predicted by the aggregation of ratings of opinion leaders and the active user's rating calculated by tag-based method. Detailed descriptions as well as the insights of the *OLrs* are given in the subsequent sections.

#### 3.1 Preliminaries

We first introduce some notations to model the recommendation problem. There are a set of users  $U$ , a set of items  $I$ , and a set of ratings  $R$ . We use the symbols  $u, v$  for the users and  $i, j$  for the items. Then  $r_{u,i}$  represents a rating given by user  $u$  on item  $i$ . At the same time, user  $u$  will tagging item  $i$  use a set  $T_u$  of tags he likes, and item  $i$  have a set  $T_i$  of tags represent its properties as well. Otherwise, in a trust-aware recommender system,  $t_{u,v} = 1$  represents user  $u$  trusted user  $v$ . And, the active user  $u$  may have some trusted neighbors, and these neighbors have some neighbors as well. Then, a trust network will construct around user  $u$ , and we can identify a set of opinion leaders in that network. Hence, the recommendation problem can be described as: given a set of user ratings  $(u, i, r_{u,i})$ , a set of user trust  $(u, v, t_{u,v})$ , and some set of tags of  $T_u$  and  $T_j$ , predict a best prediction  $(u, j, r_{u,j})$  for user  $u$  on item  $j$ .

#### 3.2 Identifying Opinion Leaders

There often are some authorities, who are the opinion leaders [26]. Comments on certain items they give can affect other people's attitude. The rating of user  $u$  on item  $i$  is affected by the ratings of opinion leaders on item  $i$  in *OLrs*. Thus, how to find the opinion leader  $k$  who has a strong influence for user  $u$  and an accurate evaluation of item  $i$  is a key work. In our presented method, the nodes of trust network are the users in recommender system. If user  $u$  trust user  $v$ , then there is an edge between these two nodes and this edge is directed from  $u$  to  $v$ . Directed trust network is constructed by the users and the trust relations between these users in recommender system.

Constructing the trust network of all users in recommender system is very exhausting. Fortunately, in order to find the opinion leaders of user  $u$ , we only need to construct the trust network of user  $u$  because the opinion leaders must be exit in this trust network.

We start from user  $u$  and use the breadth-first search algorithm to search  $d$  layers. Then the trust network of user  $u$  will be constructed by the nodes and the edges between nodes of  $d$  layers. The set of users in trust network is marked  $TN_u$ . Note that the greater  $d$  is, the more trusted neighbors will be inferred. However, the more cost will be taken and more noise is likely to be incorporated. According to the theory of six-degree separation [27], that is, any two users in

the social network can be connected (if possible) within small (less than six) steps. In this work, we restrict  $d \leq 4$  to prevent meaningless searching and save computational cost for large-scale data sets. We will discuss the influence of  $d$  in Sect. 4.

After the trust network of user  $u$  is constructed, the first step is finding out the opinion leaders who have a strong influence for user  $u$ . There are two types of influence opinion leaders in our method, one is the direct influence opinion leaders and another is the indirect influence opinion leaders. The formal definition of these influence opinion leaders is given in the following.

**Definition 3.1.** The direct influence opinion leader group of user  $u$  denoted by  $TNL_u$  is a set of users, as following:

$$TNL_u = \{v | P_v < P_{in}, t_{uv} = 1 \text{ and } v \in TN_u\} \quad (1)$$

where  $TN_u$  is a set of users who trust by the active user  $u$ , it is sorted (in descending order) by the number of each user's trustees.  $t_{uv}$  represent the relationship between the active user  $u$  and his truster  $v$ , if  $t_{uv}$  equal to 1 means  $v$  is direct trust by  $u$ .  $P_v$  is the position of truster  $v$  in  $TN_u$ , and  $P_{in}$  is a threshold be used to determine how many truster will be selected.

**Definition 3.2.** The indirect influence opinion leader group of user  $u$  denoted by  $TNG_u$  is a set of users, as following:

$$TNG_u = \{v | P_v < P_{in}, t_{uv} = 0 \text{ and } v \in TN_u\} \quad (2)$$

where  $TN_u$ ,  $t_{uv}$ ,  $P_v$  and  $P_{in}$  are as well as definition 3.1, however  $t_{uv}$  equal to 0 means  $v$  is indirect trust by  $u$ .

Because the opinion leaders have their own expertise, they can't have proper evaluation for each class of goods. For example *Bobe* is a computer science professor and he must be the opinion leader in field of computer. *Bobe* can give a fair evaluation of the books in field of computer. But if the book is in field of military, his judge on this book maybe suspect. Then, we need to identify someone in  $TNL_u$  or  $TNG_u$  for their evaluation ability on the item  $i$  which will be recommended to user  $u$ . The ratings of these opinion leaders on item  $i$  can be the important reference for user  $u$ .

Next, we use Matrix Factorization [11] and Euclidian distance to compute the evaluation ability of an opinion leader about the item  $i$ . Matrix Factorization decomposes the ratings matrix into two lower dimension matrices  $P \in R^{|U| \times d}$  and  $Q \in R^{|I| \times d}$  which contain corresponding vectors with length  $L$  for every user and item. The resulting dot product  $Q_i^T P_k$  captures the interaction between opinion leader  $k$  and item  $i$ , the leader's overall interest in the item's characteristics.

$$\hat{r}_{ki} = Q_i^T P_k \quad (3)$$

To determine the latent feature vectors, the system minimizes the regularized squared error on the set of observed ratings:

$$\min_{P^*, Q^*} \sum_{(k,i) \in R_c} (r_{k,i} - Q_i^T P_k)^2 + \lambda(\|P_k\|^2 + \|Q_i\|^2) \quad (4)$$

where  $R_c$  is the set of the  $(k, i)$  pairs for which  $r_{ki}$  is observed. Thus, Matrix Factorization characterizes every opinion leader and item by assigning them a latent feature vector  $P_k$ . We use the user feature vector to represent each opinion leader, and use the item feature vector  $Q_i$  to represent each item.

Additionally, we use the Euclidian distance to compute the evaluation ability of an opinion leader  $k$  about item  $i$ .

$$d_{ki} = \sqrt{(Q_{i1} - P_{k1})^2 + (Q_{i2} - P_{k2})^2 + \dots + (Q_{iL} - P_{kL})^2} \quad (5)$$

According to definitions 3.1 and 3.2, here we have two types of opinion leaders in our method, one is the direct opinion leaders and another is the indirect opinion leaders. The formal definition of these opinion leaders is given in the following.

**Definition 3.3.** The direct opinion leader group of user  $u$  denoted by  $OLL_u$  is a set of users, as following:

$$OLL_u = \{k | P_k < P_o, k \in TNL_u\} \quad (6)$$

where  $TNL_u$  is defined in definition 3.1 and sorted (in descending order) by  $d_{ki}$ .  $P_k$  is the position of opinion leader  $k$  in  $TNL_u$ , and  $P_o$  is a threshold be used to determine how many opinion leader will selected.

**Definition 3.4.** The indirect opinion leader group of user  $u$  denoted by  $OLG_u$  is a set of users, as following:

$$OLG_u = \{k | P_k < P_o, k \in TNG_u\} \quad (7)$$

where  $TNG_u$  is defined in definition 3.2 and sorted (in descending order) by  $d_{ki}$ .  $P_k$  and  $P_o$  are as well as definition 3.3.

After the opinion leaders who have a strong influence for the user and an accurate evaluation of item have been identified, we present details of predicting ratings in the following section.

### 3.3 Predicting

In social life when user  $u$  ratings item  $i$ , he often refer to the views of his influential people i.e. opinion leaders. But certainly not taken action totally under the opinion leader's suggestions, he will take his own judgment in the end. For this reason, the rating of user  $u$  consists of two parts in  $OLrs$ . One is  $r_{oli}$  the ratings given by the opinion leaders, and another is  $r'_{ui}$  the rating of user  $u$  himself. The rating of user  $u$  is given by tag-based algorithm. Hence,  $\hat{r}_{ui}$  is computed as a linear combination of the two parts:

$$\hat{r}_{ui} = \alpha \times r_{oli} + (1 - \alpha) \times r'_{ui} \quad (8)$$

where parameters  $\alpha$  indicate the extent to which the combination relies on opinion leaders evaluation.

The rating of user  $u$  on item  $i$  influenced by opinion leaders is related to  $T_{uk}$  the degree of be trusted of opinion leaders. In our work, there are two types of opinion leaders. According to definition 3.3 and 3.4, the trust  $T_{uk}$  user  $u$  gives to direct opinion leaders is explicit trust directly given by user  $u$ , otherwise the trust user  $u$  give to indirect opinion leaders is indirect trust and it is calculated using the following formula.

$$T_{uk} = \frac{1}{d_{u,k}} \times T'_{uk} \quad (9)$$

where  $T'_{uk}$  denotes the inferred trust value by the MoldTrust [24] algorithm,  $d_{uk}$  is the shortest distance between user  $u$  and opinion leader  $k$ , which is no more than search depth as defined in Sect. 3.2. With the increase of  $d_{uk}$ , the trust user  $u$  give to opinion leader  $k$  will decline.

Having the trusted value of user  $u$  and each of his opinion leaders, we can calculate  $r_{oli}$  as follows:

$$r_{oli} = \frac{\sum_{k \in O} T_{uk} \times r_{ki}}{\sum_{k \in O} T_{uk}} \quad (10)$$

where  $r_{ki}$  is the rating on item  $i$  given by the opinion leader  $k$ ,  $O$  is the opinion leader set  $OLL_u$  if we use the direct opinion leaders, or  $OLG_u$  when we use the indirect opinion leaders.

Inspired by [28], we will use tag-based method to compute the rating of user  $u$  on item  $i$ . Tag is a short label used to mark the item which rating by user  $u$ . If a tag is frequently used and the items marked by this tag have a high rating, it deducted that user  $u$  like using this tag to mark his favorite items. In other words, we can infer the user's rating on some item based on tags. The rating of user  $u$  on item  $i$  is calculated according to the following formula.

$$r'_{ui} = \frac{1}{|F(u, i)|} \sum_{f \in F(u, i)} \omega_u^f \quad (11)$$

where  $F(u, i)$  is a set of tag which used by user  $u$  and mark the item  $i$ ,  $|F(u, i)|$  is the number of tags in  $F(u, i)$ .  $\omega_u^f$  is the weight of tag  $f$ , and it is calculated as following.

$$\omega_u^f = \frac{1}{|u(f)|} \sum_{j \in u(f)} r_{u,j} \quad (12)$$

where  $u(f)$  is the set of item which user  $u$  used tag  $f$  to mark,  $|u(f)|$  is the number of time user  $u$  using tag  $f$ .

## 4 Evaluation

In order to verify the effectiveness of the present *OLrs* method, we conduct experiments on a real-world data set. We want to address some questions: (1) how the performance of *OLrs* in comparison with other counterparts; (2) how do the search depth  $d$  affect the recommendation results; (3) how the type of opinion leaders, direct or indirect affect the recommendation quality.

#### 4.1 Data Acquisition

We use the Epinions data set to evaluate the present method, as this data set has been widely used in previous work such as [29,30]. We obtain 47,064 ratings, assigned by 965 users on 6730 items. Each user has rated at least 2 items, and the ratings follow the 1 (bad) to 5 (excellent) numerical scale. The sparsity level of the data set is  $1 - \frac{47,064}{965 \times 6,730}$ , which is 0.9928. And there are 47,064 reviews, which can be used to extract user preference.

#### 4.2 Evaluation Metrics

To measure the predicting accuracy, we use the effective metric mean absolute error (MAE), which is defined as the average absolute difference between predicting ratings and actual ratings. It is computed by the following formula.

$$MAE = \frac{\sum_u \sum_i |r_{ui} - \hat{r}_{ui}|}{N} \quad (13)$$

where  $N$  is the number of testing ratings,  $r_{ui}$  is an actual user-specified rating on an item, and  $\hat{r}_{ui}$  is the prediction for a user on an item given by the recommender system. A lower MAE value means that the prediction is close to the ground truth. Hence, for MAE values of a recommendation algorithm, the smaller the better.

#### 4.3 Experimental Settings

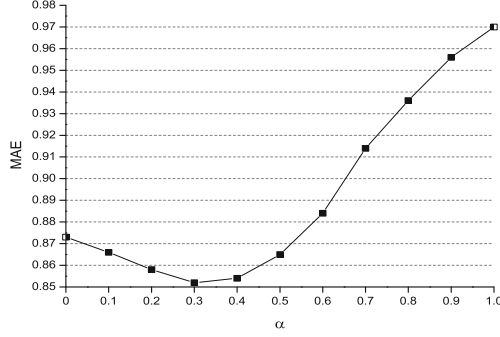
In the experiments, we compare the performance of *OLrs* with the well-known user-based CF and the TGER [28]. CF computes user similarity with Pearson Correlation Coefficient and selected the Top  $N$  neighbors whose similarity is larger. For TGER method, we adopt the settings as the author report in our experiments. Otherwise, we divide the Epinions data set into the training set and the testing set, the former consist 80 % records and the latter consist 20 % records. We obtain the recommendation predictions based on the training set and use testing set to evaluate the accuracy of recommend algorithms.

#### 4.4 Results and Analysis

**Impact of the Parameter  $\alpha$ .** As mentioned in Sect. 3, users always took the views of their opinion leaders into consideration before judge the item. How can opinion leaders affected the users attitude, it is depends on users personal features in real world. But in *OLrs*, this influence is control by the parameter  $\alpha$ . Then, we intend to determine the best  $\alpha$  for *OLrs* method. In the first experiment, we vary the parameter  $\alpha$  from 0 to 1 with step 0.1. The results are illustrated in Fig. 1.

According to Fig. 1, we found that  $\alpha$  has large impact on the recommendation accuracy. With the parameter increases from 0 to 0.3, the value of MAE decreases





**Fig. 1.** The MAE of *OLrs* with different parameter  $\alpha$

from 0.873 to 0.852. It is because the views of opinion leaders have been taken into consideration, and help user done a good judge. However, as the parameter increase dramatically, MAE become more worsens. The worst value of MAE is 0.97 when  $\alpha$  is equal to 1. This process can be explained by user's judge is not totally under the opinion leaders suggestion. We select the parameter such that the value of MAE is lower. Therefore, in the other experiments the parameter  $\alpha$  is set to 0.3.

#### Effect of the Type of Opinion Leaders and the Search Depth $D$ .

According to definitions 3.3 and 3.4, the direct opinion leaders were pike up from the users who are direct trust by the active user, while the indirect opinion leaders were identified from the users who are indirect trust by the active user. As we know that, the direct opinion leaders may be affect the active user hugely in real world, for the reason that these direct opinion leaders are the most trust person. However, it is not sure that the most trust people can make an accurate evaluation of recommended item. There are may be some indirect opinion leaders can have a good judge about the recommended item. We analyze which type of opinion leaders would have better impact on the active user in this section.

In addition, the identified of indirect opinion leaders is also related with search depth  $d$ . It is considered that trust between users is transitive [23]. By propagating trust, more trusted people will be found and they can be used to improve recommended performance. The search depth  $d$  will decide the degree of propagating trust in *OLrs*. Then, the selected value of  $d$  is very important. If this value is too small, no trusted people will be found. If this value is too large, too many opinion leaders of user  $u$  in trust network will be found and more noise is likely to be incorporated, and their impact on user  $u$  will be reduced. In this section we analyze the impact of search depth  $d$  meanwhile.

As illustrated in Table 1, the value of MAE of *OLrs* based on direct opinion leaders is all 0.852. It is show that the search depth  $d$  affects nothing on direct opinion leaders, because this type of opinion leaders is directly trusted by the active user. While the search depth  $d$  affects the MAE of the algorithm based

**Table 1.** The MAE of OLrs with different type of leaders

	Type of leaders	search depth $d$			
		1	2	3	4
MAE	direct opinion leaders	0.852	0.852	0.852	0.852
	indirect opinion leaders	0.852	0.852	0.851	0.854

on indirect opinion leaders. We can found that the best value of MAE is 0.851 if the search depth  $d$  is 3. Thus, *OLrs* have a better performance if the indirect opinion leaders be used.

**Comparison with State-of-the-Art Methods.** Table 2 shows the MAE of *OLrs* and other compared methods. According to these values, we can found that *OLrs* is the best method for its MAE is 0.851, and CF is the worst method. The MAE of CF indicates that the data sparsity problem has been deteriorated the recommend accuracy, and we need some additional information beside user-item ratings to make improvement. TGER is the method which uses Tag to enhance recommends performance. The MAE of TGER is 0.873 which better than CF. Furthermore, *OLrs* achieves best performance than other methods by taking into account opinion leaders and user preference.

**Table 2.** The MAE of *OLrs* and other methods

method	OLrs	TGRE	CF
MAE	0.851	0.873	1.42

## 5 Conclusions and Future Works

To address the data sparsity problem, in this paper we have proposed *OLrs*, a recommend algorithm that combines opinion leaders and user preference in a unified frame. Experimental results based on the real-world data set demonstrate that our method outperforms other counterparts in terms of accuracy. In conclusion, we proposed an effective method to improve the performance of recommender systems. However, the present work depends on explicit trust which maybe not so easy to extract from social networks due to the privacy. Hence, one possible future work is to use the implicit trust in *OLrs*.

**Acknowledgments.** The authors are grateful to the anonymous reviewers and the helpful suggestion given by the partners. The research was supported by the National Natural Science Foundation of China (no. 61300137), the Foundation for Distinguished Young Teachers in Higher Education of Guangdong (no. Yq2014117), the Technology Project of Zhanjiang (no. 2013B01148), the Natural Science Foundation of Lingnan Normal College (no. QL1307, no. QL1410).

## References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
2. Huang, Z., Chen, H., Zeng, D.: Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst. (TOIS)* **22**(1), 116–142 (2004)
3. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. *ACM Comput. Surv. (CSUR)* **47**(1), 3 (2014)
4. Moshfeghi, Y., Piwowarski, B., Jose, J.M.: Handling data sparsity in collaborative filtering using emotion and semantic based features. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 625–634. ACM (2011)
5. Robu, V., Halpin, H., Shepherd, H.: Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Trans. Web (TWEB)* **3**(4), 14 (2009)
6. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Signed networks in social media. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1361–1370. ACM (2010)
7. Summers, J.O.: The identity of women’s clothing fashion opinion leaders. *J. Mark. Res.* **7**, 178–185 (1970)
8. Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: *IEEE 10th International Conference on Data Mining (ICDM)*, pp. 176–185. IEEE (2010)
9. Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al.: The youtube video recommendation system. In: *Proceedings of the fourth ACM Conference on Recommender Systems*, pp. 293–296. ACM (2010)
10. Böhmer, M., Hecht, B., Schöning, J., Krüger, A., Bauer, G.: Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 47–56. ACM (2011)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
12. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: *Advances in Neural Information Processing Systems*, pp. 1257–1264 (2007)
13. Singh, A.P., Gordon, G.J.: Relational learning via collective matrix factorization. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658. ACM (2008)
14. Xie, H., Li, Q., Mao, X., Li, X., Cai, Y., Zheng, Q.: Mining latent user community for tag-based and content-based search in social media. *Comput. J.* **57**(9), 1415–1430 (2014)
15. Sen, S., Vig, J., Riedl, J.: Tagommenders: connecting users to items through tags. In: *Proceedings of the 18th International Conference on World Wide Web*, pp. 671–680. ACM (2009)
16. Xie, H.R., Li, Q., Cai, Y.: Community-aware resource profiling for personalized search in folksonomy. *J. Comput. Sci. Technol.* **27**(3), 599–610 (2012)
17. Cai, Y., Li, Q., Xie, H., Min, H.: Exploring personalized searches using tag-based user profiles and resource profiles in folksonomy. *Neural Netw.* **58**, 98–110 (2014)

18. Zhen, Y., Li, W.J., Yeung, D.Y.: Tagicofi: tag informed collaborative filtering. In: *Proceedings of the third ACM Conference on Recommender Systems*, pp. 69–76. ACM (2009)
19. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: *Proceedings of the 10th International Conference on Intelligent user Interfaces*, pp. 167–174. ACM (2005)
20. Seth, A., Zhang, J., Cohen, R.: Bayesian credibility modeling for personalized recommendation in participatory media. In: De Bra, P., Kobsa, A., Chin, D. (eds.) *UMAP 2010. LNCS*, vol. 6075, pp. 279–290. Springer, Heidelberg (2010)
21. Ray, S., Mahanti, A.: Improving prediction accuracy in trust-aware recommender systems. In: *2010 43rd Hawaii International Conference on System Sciences (HICSS)*, pp. 1–9. IEEE (2010)
22. Chowdhury, M., Thomo, A., Wadge, W.W.: Trust-based infinitesimals for enhanced collaborative filtering. In: *COMAD* (2009)
23. Golbeck, J.A.: *Computing and applying trust in web-based social networks* (2005)
24. Massa, P., Avesani, P.: Trust-aware recommender systems. In: *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 17–24. ACM (2007)
25. Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397–406. ACM (2009)
26. Watts, D.J., Dodds, P.S.: Influentials, networks, and public opinion formation. *J. Consum. Res.* **34**(4), 441–458 (2007)
27. Watts, D.J.: *Six Degrees: The Science of a Connected Age*. WW Norton and Company, New York (2004)
28. Cai Yi, Liu Yu, Z.G.C.J.M.H.: Tag group effect-based recommendation algorithm for collaborative tagging systems. *Journal of South China University of Technology (Natural Science Edition)* **41**(9), 65–70 (2013)
29. Guo, G., Zhang, J., Thalmann, D., Basu, A., Yorke-Smith, N.: From ratings to trust: an empirical study of implicit trust in recommender systems. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 248–253. ACM (2014)
30. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 2747–2753. AAAI Press (2013)

Database Systems for Advanced Applications

DASFAA 2015 International Workshops, SeCoP, BDMS,  
and Posters, Hanoi, Vietnam, April 20-23, 2015, Revised  
Selected Papers

Liu, A.; Ishikawa, Y.; Qian, T.; Nutanong, S.; Cheema,  
M.A. (Eds.)

2015, XIV, 328 p. 99 illus., Softcover

ISBN: 978-3-319-22323-0