Received September 29, 2018, accepted October 14, 2018, date of publication October 24, 2018,
date of current version November 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877208

A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks

RUI CHEN[1,2], QINGYI HUA[1], YAN-SHUO CHANG[3], BO WANG[1,4], LEI ZHANG[5], AND XIANGJIE KONG[6], (Senior Member, IEEE)
[1]School of Information Science and Technology, Northwest University, Xi'an 710127, China
[2]School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou 450015, China
[3]Insititute for Silk Road Research, Xi'an University of Finance and Economics, Xi'an 710100, China
[4]School of Computer, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[5]Department of Public Computer Teaching, Yuncheng University, Yuncheng, 044000, China
[6]Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, School of Software, Dalian University of Technology, Dalian 116620, China

Corresponding authors: Qingyi Hua (nwuchenrui@126.com and huaqy@nwu.edu.cn), Yan-Shuo Chang (396336805@qq.com), and Xiangjie Kong (xjkong@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61272286, in part by the joint funded projects of the Special Scientific Research Fund for Doctoral Program of Higher Education under Grant 20126101110006, in part by the Industrial Science and Technology Research Project of Shaanxi Province under Grant 2016GY-123, in part by the Blue Book of Science Research Report on the Belt and Road Tourism Development under Grant 2017sz01, and in part by the Research Foundation of Xi'an University of Finance and Economics under Grant 17FCZD02.

ABSTRACT In the era of big data, recommender system (RS) has become an effective information filtering tool that alleviates information overload for Web users. Collaborative filtering (CF), as one of the most successful recommendation techniques, has been widely studied by various research institutions and industries and has been applied in practice. CF makes recommendations for the current active user using lots of users' historical rating information without analyzing the content of the information resource. However, in recent years, data sparsity and high dimensionality brought by big data have negatively affected the efficiency of the traditional CF-based recommendation approaches. In CF, the context information, such as time information and trust relationships among the friends, is introduced into RS to construct a training model to further improve the recommendation accuracy and user's satisfaction, and therefore, a variety of hybrid CF-based recommendation algorithms have emerged. In this paper, we mainly review and summarize the traditional CF-based approaches and techniques used in RS and study some recent hybrid CF-based recommendation approaches and techniques, including the latest hybrid memory-based and model-based CF recommendation algorithms. Finally, we discuss the potential impact that may improve the RS and future direction. In this paper, we aim at introducing the recent hybrid CF-based recommendation techniques fusing social networks to solve data sparsity and high dimensionality and provide a novel point of view to improve the performance of RS, thereby presenting a useful resource in the state-of-the-art research result for future researchers.

INDEX TERMS Recommender systems, collaborative filtering, matrix factorization, singular value decomposition, trust-aware collaborative filtering, social networks.

I. INTRODUCTION

With the rapid expansion of Internet technology and ubiquitous computing, a variety of channels and methods to access to information have brought great convenience for users. However, the geometric growth of data makes it difficult for users to find information that meets their own needs in time, so "big data" leads to "information overload" problem, and makes a lot of irrelevant redundant information interfere with users' choice [1], [2]. In the era of big data, RS does not require users to provide clear needs, and establish users' interest models by analyzing their historical behavior to recommend items which better match the active users' interests [7], [8].

Collaborative Filtering (CF) is one of the most widely used and successful technologies in RS. CF-based recommendation techniques have achieved great success, and have a wide range of application prospects in many fields such as e-commerce and social networks. However, as big data arise, the CF-based approach often suffers from several shortcomings [51], such as data sparsity, cold start, and scalability issues, which seriously affect the recommended quality of RS. To tackle the aforementioned problems, many data mining and machine learning techniques such as clustering [27], [29], singular value decomposition (SVD) [11], [39], probability matrix factorization (PMF) [64], [87], [88], and non-negative matrix factorization (NMF) [47], [75], [76] are proposed to improve the performance of RS. To solve the problems of data sparsity and cold start in the era of big data, social factors are recently considered to further improve the performance of RS [50], [57], [74], [75], [78], [82], [86], [88], [89], such as reliability-based trust-aware collaborative filtering(RTCF) [32], recommendation with social trust ensemble (RSTE) [70], a matrix factorization based model for recommendation in social rating networks (SocialMF) [81], a state-of-art social network-based recommender system (SNRS) [89], an enhanced personalized recommendation model based on user attributes clustering and rating filling (EPRM) [88], and a neighborhood-aware unified probabilistic matrix factorization (NAUPMF) [87].

A. PRIOR RELATED SURVEYS

In the past few years, some survey or review articles have been presented in RS. A number of studies review system frameworks, overview, and methods of RS from a methodological point of view [1], [9], [12], [15]. For instance, Wang *et al.* [1] outline system frameworks, main models, key frameworks, assessments and typical applications of context-aware RS with a process-oriented view. Adomavicius and Tuzhilin [12] present an overview of the field of RS and describe the current recommendation methods: content-based, CF-based, hybrid recommendation approaches. Yang *et al.* [17] propose a framework of CF-based RS according to a variety of users' data including ratings from users and user historical behavior, and compares several typical CF algorithms. Most of the existing review articles discuss traditional methods and techniques of RS, a few of which involve social recommendation methods [3], [7], [36], [63], [73]. For instance, Lü *et al.* [3] review recent progress of RS and discusses the major challenges, such as dimensionality reduction techniques, similarity-based approaches, and social filtering. Tang *et al.* [73] present a review of existing RS, give the definitions of social recommendation, and discuss the feature of social recommendation and its implications. Yang *et al.* [36] provide a brief overview over the task of RS and traditional approaches, and present how social network information can be adopted by RS. Although some review studies have referred to social recommendation methods [3], [10], [36], [42], [73], they don't systematically introduce social networks-based recommendation methods for dealing with data sparsity and cold start problems, and some of the social factors have not been fully considered [61], [70], [74], [79], [81]–[83], [89].

B. CONTRIBUTIONS OF THIS SURVEY

This paper is a systematical survey that provides a comprehensive review of existing work on conventional CF-based and hybrid CF-based recommendation methods. Our major contributions can be summarized as follows:

- We mainly summarize the traditional and hybrid model-based CF recommendation methods, techniques and new research progress on RS for providing some references and research inspiration for the future research.
- We survey the social networks-based recommendation methods in recent years, and present recent studies on CF-based recommendation algorithms to solve the problems of data sparsity and cold start.
- We study numerous influences of social factors on the recommendation quality of RS.
- We discuss several potential issues of CF and highlight future research directions for solving the problems of data sparsity and cold start.

The remainder of this review is organized as follows. In Section 2 we present an overview of RS and review the frequently used CF approaches, techniques, evaluation metrics, and technical challenges on existing methods. Next, in Section 3 we introduce the techniques and modeling approaches used in the hybrid CF-based recommender systems, such as enhanced similarity measures, memory-based trust-aware CF, model-based social matrix factorization-based CF, and reduce dimensionality. Then we discuss the advantages of CF in Section 4. Finally, we outline conclusions, prospect of further study and development trends in Section 5.

II. RECOMMENDER SYSTEMS

A. OVERVIEW

A complete RS consists of the following three parts: user, item resource and recommendation algorithm, which is as shown in Fig. 1. The user model is established by analyzing the users' interests and preferences, likewise, the model for item resource is established according to items' feature. Then, the characteristics of the user are compared with the characteristics of all items to predict which items the user might like by using the recommendation algorithm, and the predicted results are recommended to the user. Among them, the recommendation algorithm is the most important part of RS [18], [41]. The performance of the proposed algorithm directly affects the overall performance of the RS. Therefore, the research work of RS is mainly focused on the design and implementation of the proposed algorithm.

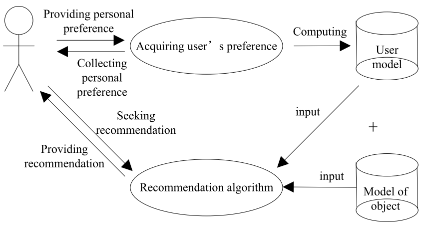


FIGURE 1. A model of recommender systems.

In general, RS has been divided into several different categories, namely CF, content-based, social filtering, association rule mining, and social filtering [14].

B. APPLICATIONS

The task of RS is to convert users' historical behavioral information on items into predictions of users' possible future interests and preferences, and help users find items (movies, music, books, Web information, etc) that may be interested in from a large amount of data by mining the binary relation between users and items [3]. After its first appearanceĄCRS attracts more and more attention and has been widely applied in industrial communities such as digital information content services, e-commerce, information retrieval, mobile news, e-tourism, education, digital libraries and so on. The recommendation for many e-commerce sites is based on CF algorithm. For instance, Amazon's 20% -40% of sales is due to RS, and 60% of DVDs rented by Netflix are selected based on RS [3], [8]–[10].

Table 1 shows main recommendation systems that are being used in various fields.

TABLE 1. The applications of RS in various fields.

Category	Recommender systems
Video	Netflix, Hulu, Youtube, Youku, iQiyi, Tudou, Ku6
News	Google news, ifeng, Toutiao, NetEase news, Digg
Music	Yahoo! Music, Pandora, Douban music, QQ music, Google play, Last.fm
Socia networking services	Facebook, Twitter, sina weibo, QQ, Linkedin
E-business	Amazon, eBay, taobao, JD, Suning

C. TRADITIONAL CF RECOMMENDATION METHODS

In this section, we will introduce the most commonly used CF-based recommendation methods, including latent factor model (LFM), and its existing variations such as matrix factorization, NMF, and SVD. Traditional CF can be divided into the two methods: memory-based and model-based methods. The framework of CF-based RS is shown in Fig. 2.

1) MEMORY-BASED CF TECHNOLOGY

Memory-based CF recommendation algorithm obtains the similar relationships between users or items according to the user-item rating matrix, and then recommends the items that are highly rated by similar users for the active user [17]. In memory-based CF, the ratings on items from users are

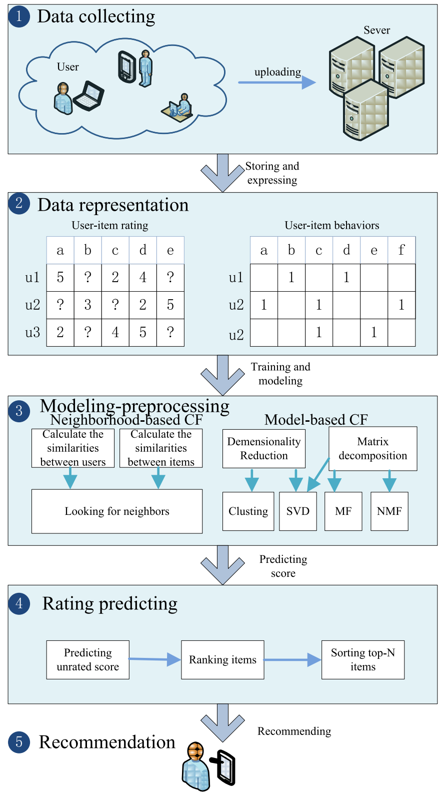


FIGURE 2. The framework of collaborative filtering-based RS.

directly used to predict unknown ratings for new items. The memory-based recommendation method can be subdivided into two ways: user-based CF and item-based CF [2]. The rationale of user-based CF and item-based CF is shown in Fig. 3.

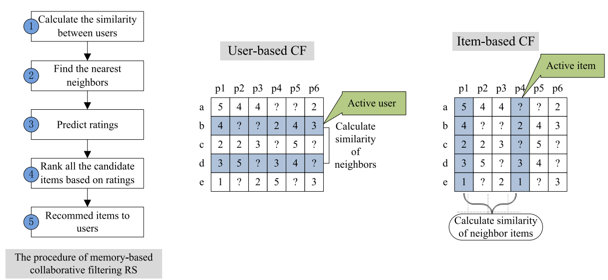


FIGURE 3. The rationale of user-based CF and item-based CF.

a: USER-BASED CF RECOMMENDATION ALGORITHM

The idea of user-based CF is that users with similar historical ratings should have similar interests, so we can predict the active user's missing ratings on the specific items according to similar users' ratings on given items. Firstly, the similarities between the active user and other users are calculated, and then the neighbors of the active user are selected according to the similarities. Finally, the ratings from the active user are predicted according to the historical preference information of the similar neighbor users, and the recommendation results are generated [12], [18].

(1) Calculate the Similarity between Users. The ratings of the user u are usually expressed as the rating vector $r_u = \{r_{u1}, r_{u2}, \ldots, r_{un}\}$. The similarity between the two users is obtained by comparing the rating vectors of the two users. The classical measures to calculate the similarity between users are cosine similarity and Pearson correlation coefficient (PCC).

Cosine similarity: the user's ratings can be indicated as an n-dimensional vector, and the similarity between users is obtained through the user's rating vector angle. In general, the smaller the angle is, the higher the similarity is. Cosine vector similarity is calculated as follows [2] (see Eq. 1):

$$\begin{aligned} sim_{uv} &= \cos(\vec{r}_u, \vec{r}_v) = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\|_2 \times \|\vec{r}_v\|_2} \\ &= \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2}\sqrt{\sum_{i \in I_v} r_{vi}^2}} \end{aligned} \tag{1}$$

where sim_{uv} represents the similarity between users u and v, $\vec{r}_u$ and $\vec{r}_v$ represent the rating vectors of u and v, respectively, $\|\vec{r}_u\|_2$and $\|\vec{r}_v\|_2$ represent 2-norm of u and v, respectively, and r_{ui} and r_{vi} represent the ratings of u and v on the item i, respectively. I_u and I_v represent the sets of items rated by users u and v, respectively, and I_{uv} represents the set of items commonly rated by both u and v.

Pearson correlation coefficient is calculated as follows [51] (see Eq. 2):

$$sim_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)}\sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)}} \tag{2}$$

where $\bar{r}_u$and $\bar{r}_v$ represent the average ratings from u and v, respectively.

(2) Find the Nearest Neighbors. There are usually two methods for finding nearest neighbors: k-nearest neighbors and setting threshold. k-nearest neighbors method is to select the first k users with the closest similarity to the active user u as his or her nearest neighbors. The threshold method means that a threshold δ is set initially, when the similarity between user v and the active user u is greater than δ, the user v is selected as one of the nearest neighbors.

(3) Predict Ratings. There are two main ways to make recommendations for an active user: predicting the ratings and providing a top-N recommendation list. The both need to predict ratings of the active user u on a new item i using the ratings on i from users most similar to u. The predicted rating is calculated as follows [32] (see Eq. 3):

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} sim_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in N_u} |sim_{uv}|} \tag{3}$$

where N_u denotes the similar neighbor set of the user u.

Top-N recommendation is mainly used in the following scenarios: shopping websites or websites that generally do not have explicit rating information. In this case, through the user's implicit feedback information, a list of items that may be of interest is recommended to the user, and some useful data is extracted to form a user-item matrix where each element is 0 or 1 [17]. In general, the user's preferences are modeled in point-wise way, each user's rating for each item (or a probability value between 0 and 1) is predicted, and then the rated items are sorted in descending order, finally top-N items are recommended to users. Memory-based CF for binary data can actually be considered as a special case of memory-based CF for ratings. The rating $r_{ui} = 1$ if the user-item pair (u,i) is observed, and $r_{ui} = 0$ otherwise in the feedback matrix R. Therefore, the cosine vector similarity for binary ratings is calculated as follows [107] (see Eq. 4):

$$\begin{aligned} sim_{uv} = \cos(\vec{r}_u, \vec{r}_v) &= \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2}\sqrt{\sum_{i \in I_v} r_{vi}^2}} \\ &= \frac{|I_u \cap I_v|}{\sqrt{|I_u|}\sqrt{|I_v|}} \end{aligned} \tag{4}$$

where I_u and I_v denote the sets of items observed by users u and v, respectively, and I_{uv} denotes the set of items commonly observed by both u and v.

Rating predictions for binary data can be calculated using Eq.(3) as well. Unlike the recommendation method for rating prediction, the value of the predicted rating $\hat{r}_{ui}$ in implicit feedback scenarios will be a rating of between 0 and 1. For top-N recommendation, RS recommends the first n items by ranking all the items according to their predicted ratings in descending order [17], [107].

b: ITEM-BASED CF RECOMMENDATION ALGORITHM

Similar to the user-based CF recommendation algorithm, the item-based CF recommendation algorithm is also executed in the following three steps: (1) Calculate the similarity between items according to the user-item rating matrix; (2) Select the similar neighbor items according to the similarity; (3) Predict unknown ratings on the active item according to the neighbor items, and generate a recommended list.

(1) Calculate the Similarity between Items. The classical measures between items are adjusted cosine vector and Pearson correlation coefficient.

(a) Adjusted cosine vector. The adjusted cosine vector method is calculated as follows [2] (see Eq. 5):

$$sim_{ij} = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_u)\left(r_{uj} - \bar{r}_u\right)}{\sqrt{\sum_{u \in U_i} (r_{ui} - \bar{r}_u)^2}\sqrt{\sum_{u \in U_j} (r_{uj} - \bar{r}_u)^2}} \tag{5}$$

where sim_{ij} denotes the similarity between items i and j. U_i and U_j represent the sets of users who rated items i and j, respectively, and U_{ij} denotes the set of users who rated both items i and j.

(b) Pearson correlation coefficient method is calculated as follows [14] (see Eq. 6):

$$sim_{ij} = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)\left(r_{uj} - \bar{r}_j\right)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_u)^2}\sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}} \tag{6}$$

where $\bar{r}_i$ and $\bar{r}_j$ represent the average ratings on i and j in U_{ij}, respectively.

(2) Find the Nearest Neighbors. Similar to the user-based CF, there are usually two methods for finding the nearest neighbors in the item-based CF recommendation methods: k-nearest neighbors and setting threshold.

(3) Predict Ratings [69] (see Eq. 7):

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in N_i} sim_{ij} \times (r_{uj} - \bar{r}_j)}{\sum_{j \in N_i} |sim_{ij}|} \tag{7}$$

where N_i is the similar neighbor set of the item i.

2) MODEL-BASED CF TECHNOLOGY

The memory-based RS is simple to implement and the algorithms is easy to understand. However, the memory-based RS is not suitable for practical applications when dealing with large amounts of users and items. In this case, the model-based RS emerge subsequently, which can avoid the important drawback [76]. The model-based RS requires a learning phase in advance for finding out the optimal model parameters before making a recommendation. Once the learning phase is finished, the model-based RS can predict the ratings of users very quickly. Among them, latent factor model (LFM) is very competitive and widely adopted to implement RS, which factorizes the user-item rating matrix into two low-rank matrices: the user feature and item feature matrices. It can alleviate data sparsity using dimensionality reduction techniques and usually produce more accurate recommendations than the memory-based CF approach, while drastically decreases the memory requirement and computation complexity [3], [44]. SVD [11], [20], matrix factorization (MF) [21], [80], and NMF [47], [49] are usually used recommendation methods, which all take advantage of LFM.

a: MATRIX FACTORIZATION MODEL

The recommendation procedures of RS based on MF model is shown in Fig. 4.

(1) Construct Latent Feature Model. The matrix factorization model is described as follows [3] (see Eq. 8):

$$R \approx PQ^T \tag{8}$$

For the user-item rating matrix R of $m \times n$, the MF model represents approximately R as a product form of users' feature matrix P of $m \times k$ and items' feature matrix Q of $n \times k$ according to MF technique. Here, m and n are the numbers of users and items, respectively, and k is the number of latent features. An example of MF on movie recommendation is shown in Fig. 5.

(2) Obtain the Objective Function. The MF model usually aim at minimizing deviation between the decomposition of the approximate matrix and the original user-item rating matrix. Therefore, we train the model by using the gradient descent method to achieve the optimal solution [2], [3]. The objective function is described as follows [3], [18] (see Eq. 9):

$$L = \min \|R - \widehat{R}\| = \min(\sum (r_{ui} - p_{uk}q_{ki})^2 + \lambda_p \|p_{uk}\|^2 + \lambda_q \|q_{ki}\|^2 \tag{9}$$

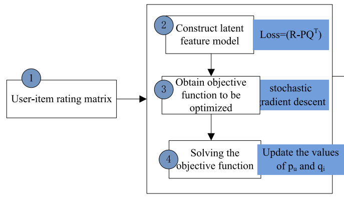


FIGURE 4. The recommendation procedure of RS based on MF.

where r_{ui} denotes the rating of user u on item i in the original matrix, and p_{uk} and q_{ki} denote the kth feature from user u and kth feature from item i in P and Q^T, respectively. λ_p and λ_q are the regularized term parameter to avoid overfitting.

(3) Update the Values of the Feature Matrices P and Q. In RS, stochastic gradient descent (SGD) [21] and alternating least squares (ALS) are often used to solve the parameters of the above objective function. SGD continuously updates the unknown parameters p_{uk} and q_{ki} until convergence according to the gradient descent direction of the objective function [27]. In order to solve Eq. (9), p_u and q_i are initialized randomly at first, and then the prediction error between the true rating and the predicted rating is calculated as follows [2], [34] (see Eq.10):

$$e_{ui} = r_{ui} - p_{uk}q_{ki} \tag{10}$$

The values of p_u and q_i are updated to obtain the approximate values using SGD method, which can be described as follows (see Eq. 11-12):

$$p_{uk} \leftarrow p_{uk} + \eta(q_{ki} \cdot e_{ui} - \lambda_p p_{uk}) \tag{11}$$
$$q_{ki} \leftarrow q_{ki} + \eta(p_{uk} \cdot e_{ui} - \lambda_q q_{ki}) \tag{12}$$

where η indicates the learning rate. The derivation process is as follows (see Eq. 13-14).

$$p_{uk} \leftarrow p_{uk} - \eta \frac{\partial L}{\partial p_{uk}} = p_{uk} - \frac{\partial}{\partial p_{uk}}(r_{ui} - \hat{r}_{ui})^2 \tag{13}$$
$$q_{ki} \leftarrow q_{ki} - \eta \frac{\partial L}{\partial q_{ki}} = q_{ki} - \frac{\partial}{\partial q_{ki}}(r_{ui} - \hat{r}_{ui})^2 \tag{14}$$

(4) Predict the Unknown Ratings according to the Matrices P and Q. The unknown ratings can be predicted as follows [3] (see Eq. 15):

$$\hat{r}_{ui} = \sum_{k=1}^{K} p_{uk}q_{ki} \tag{15}$$

For binary data, it is possible to make a prediction using the above method by assuming that R=1 for all observed user-item pairs in implicit feedback scenarios. Therefore, the objective function for binary data is described as follows [107] (see Eq. 16):

$$\begin{aligned} L &= \min_{(u,i)\in D} \|1 - \widehat{R}\| \\ &= \min \sum_{(u,i)\in D} ((1 - p_{uk}q_{ki})^2 + \lambda_p \|p_{uk}\|^2 + \lambda_q \|q_{ki}\|^2) \end{aligned} \tag{16}$$

Here, the predicted rating $\hat{r}_{ui}$ can be calculated as Eq.(15) as well, which represents a user's preference level for an item.

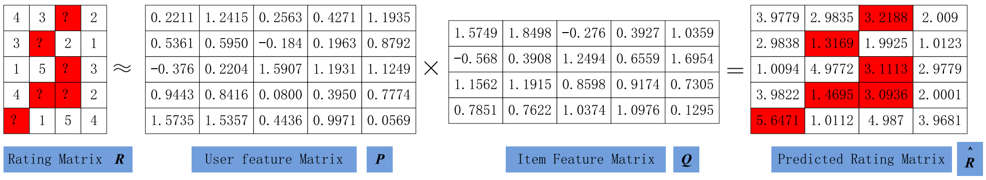


FIGURE 5. An example of matrix factorization on movie recommendation.

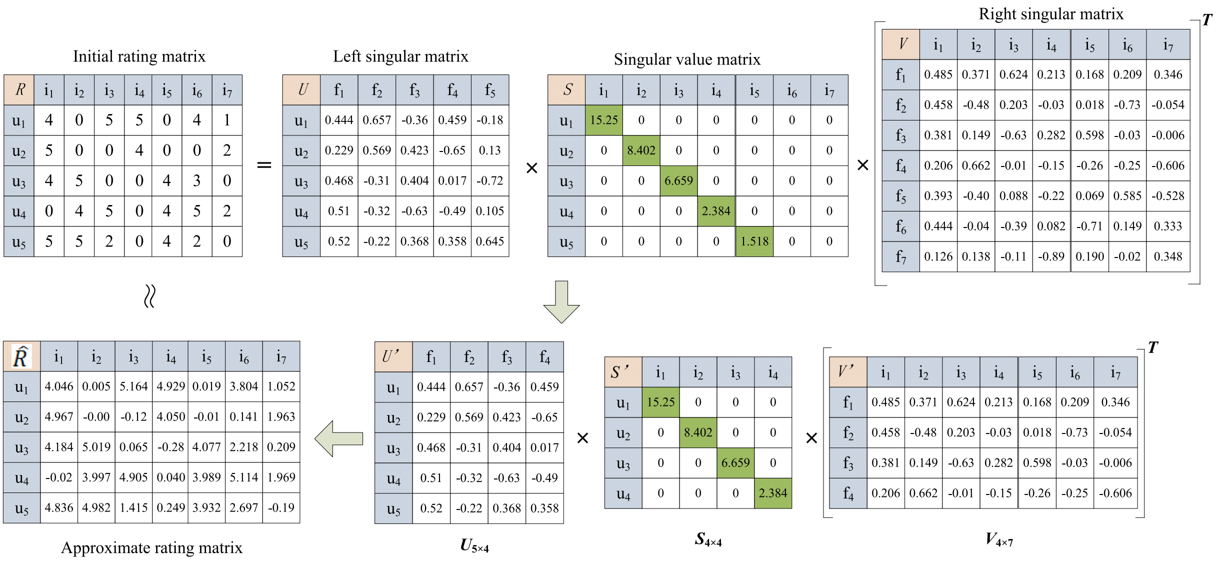


FIGURE 6. A matrix decomposition process of SVD.

b: NON-NEGATIVE MATRIX FACTORIZATION MODEL

Similarly, NMF also factorizes the original user-item rating matrix R into two matrices P and Q with rank r, where P is equal to $|U|\times f$, Q is equal to $f\times|I|$ and $f\ll\min(|U|, |I|)$. Note that a decomposition process is performed under the non-negative constraint, i.e., $P\geq 0, Q\geq 0$. Therefore, the problem of NMF-based CF is described as follows (see Eq.17) [75].

$$\arg\min loss = \|R-PQ\|^2, \quad \text{s.t. } P, Q \geq 0 \tag{17}$$

To make sure that the P and Q are non-negative, the learning rates are manipulated as follows [30], [47], [75] (see Eq.18):

$$\alpha_{uk} = \frac{p_{uk}}{(PQQ^T)_{uk}}, \quad \alpha_{ki} = \frac{q_{ki}}{(P^TPQ)_{ki}} \tag{18}$$

The updating process is described as follows [49], [75] (see Eq.19):

$$p_{uk} \leftarrow p_{uk}\frac{(RQ^T)_{uk}}{(PQQ^T)_{uk}}, \quad q_{ki} \leftarrow q_{ki}\frac{(P^TR)_{uk}}{(P^TPQ)_{uk}} \tag{19}$$

c: SINGULAR VALUE DECOMPOSITION (SVD)

Data sparsity and high dimensionality are recurring problems in RS. Therefore, dimensionality reduction is an urgent problem to be solved at present, and SVD namely a particular realization of the MF algorithms, is a powerful technique for dimensionality reduction [2]. An original rating matrix $R_{m\times n}$ can be decomposed into U, S and V according to SVD technology as follows (see Eq. 20):

$$R_{m\times n} = U_{m\times m}S_{m\times n}V^T_{n\times n} \tag{20}$$

where $U^TU = I_{m\times m}$, and $V^TV = I_{n\times n}$. Each column of U is called a left singular vector, S is a diagonal matrix, and the diagonal values are arranged from large to small, which are called singular values; each row of V^T is called the right singular vector. The value of the diagonal on the matrix S is indeed the square root of RR^T or R^TR. For instance, a matrix decomposition process of SVD is shown in Fig. 6.

As shown in Fig. 6, the dimension of the initial matrix R is reduced, which is represented by using U, S, and V. Among them, U reflects the user information, V reflects the item information, and S reflects the importance of the feature. We select the first 4 features, which take up more than 95% of the original energy. Finally, $\widehat{R}$ approximates to the real matrix R.

In general, S is a $k\times k$ diagonal matrix, where $k=\min(m,n)$. R is approximated with $\widehat{R}$ given by $R\approx\widehat{R} = U\hat{\Sigma}V$, and $\hat{\Sigma}$ is the k-rank approximation of Σ.

D. EVALUATION METRICS

Several metrics are used to evaluate the efficiency such as accuracy, coverage, and diversity in RS.

The mean absolute error (MAE) is a widely used metric to calculate the recommender's prediction [69]. MAE is calculated using the following expression (see Eq. 21):

$$\mathrm{MAE} = \frac{\sum_{(u,i)\in T} \left| r_{ui} - \hat{r}_{ui} \right|}{|T|} \tag{21}$$

where T denotes an item set. For a given RS, the lower the MAE is, the higher the prediction is, and the better the performance of the algorithm is [30].

Similar to MAE, the root mean squared error (RMSE) is also a frequently employed metric, which evaluates the absolute difference between the observed and predicted ratings as follows [30] (see Eq. 22):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{(u,i)\in T} \left(r_{ui} - \hat{r}_{ui} \right)^2}{|T|}} \tag{22}$$

In addition, the precision@N (P@N) and recall@N (R@N) [45], [62] are used to measure the recommendation accuracy by calculating the ratio of the predicted rating to the actual rating in the entire test set. The higher the precision is, the better the recommendation accuracy is. P@N and R@N are described as follows (see Eq. 23-24).

$$\mathrm{P@N} = \frac{|\text{items\_relevanted} \cap \text{topN\_items}|}{|N|} \tag{23}$$

$$\mathrm{R@N} = \frac{|\text{items\_relevanted} \cap \text{topN\_items}|}{|\text{items\_relevanted}|} \tag{24}$$

where items_relevanted and topN_items denote the actually visited list and the recommended list, respectively.

The accuracy of recommendation is also evaluated by the precision/recall. The precision describes how many percentages of the final recommended list is in user-item rating records that have taken place, and the recall describes how many percentages of user-item rating records are included in the final recommended list. The precision and recall are described as follows (see Eq. 25-26):

$$\mathrm{Precision} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |R(u)|} \tag{25}$$

$$\mathrm{Recall} = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u |T(u)|} \tag{26}$$

where R(u) denotes the number of items recommended to the user u, and T(u) denotes the user u likes the collection of items on the test set.

Area under curve (AUC) is also used to evaluate the quality of recommendation, which is described as follows [106], [110] (see Eq. 27):

$$\mathrm{AUC} = \frac{1}{|U|} \sum_{u\in U} \frac{1}{|E(u)|} \sum_{(i,j)\in E(u)} \varphi(\hat{r}_{ui} > \hat{r}_{uj}) \tag{27}$$

where |U| represents the total number of users in the test set. $E(u) = \{(i,j) | i \in T(u), j \notin T(u)\}$, and T(u) denotes the set of items on which user u performs target action. In the test set T(u), $\varphi(x)$ denotes the indicator function that is equal to 1 if x is true, and equal to 0 otherwise.

The overall prediction accuracy of the algorithm can be evaluated by mean average precision (MAP), which is the mean of the average precision (AP) of all test users. Given a user u_i and his/her sorted recommendation list $< j_1, j_2, \ldots, j_M >$ of length M, and the selected N items, AP can be calculated as follows [106], [112], [124] (see Eq. 28):

$$\mathrm{AP}_i = \frac{\sum_{k=1}^{M} \mathrm{precision}(k) \times \mathrm{ref}(k)}{N} \tag{28}$$

where precision(k) is the accuracy of top-k. If j_k hits, then ref(k)=1; otherwise, ref(k)=0. The higher the MAP is, the higher the recommendation accuracy of the algorithm is.

Another indicator of recommendation accuracy is mean reciprocal rank (MRR), namely the mean of reciprocal of the user's actual response in the recommended list. MRR is defined as follows [112], [124] (see Eq. 29):

$$\mathrm{MRR} = \frac{1}{|U|} \sum_{u\in U} \frac{1}{\min_{i\in T(u)} p} \tag{29}$$

where p indicates the rank in the recommended list.

In addition, the coverage rate is also used to evaluate the performance of RS. The coverage rate reflects the ability of the recommendation algorithm to discover the long tail. The higher the coverage rate is, the more the recommendation algorithm can recommend items in the long tail to users. The coverage is described as follows [34] (see Eq. 30):

$$\mathrm{Coverage} = \frac{|\cup R(u)|}{|I|} \tag{30}$$

The coverage indicates what percentage of the final recommended list contains items. The RS with 100% coverage can recommend each item to at least one user. Top rankings have a low recommended coverage and will only recommend popular items that account for a small percentage of the total items. A good RS not only needs higher user satisfaction but also higher coverage.

In order to satisfy users' extensive interests, the recommended list needs to cover different areas of interest of users, i.e., the recommended results need to be diversified. Diversity describes the dissimilarity between the two items in the list. Assuming s(i, j) defines the similarity between the items i and j, then the diversity of user u's recommended list is defined as follows [45] (see Eq. 31):

$$\mathrm{Diversity}\,(R\,(u)) = 1 - \frac{\sum_{i,j\in R(u)} s(i,j)}{\frac{1}{2}\,|R\,(u)|\,(|R\,(u)|-1)} \tag{31}$$

E. EXPERIMENTAL DATASETS

The real life experimental datasets used in RS can be divided into two categories: the datasets with trust relationships and the datasets without trust relationships:

1) THE DATASETS WITHOUT TRUST RELATIONSHIPS

(1) Movielens dataset is collected by the GroupLens research project team of the University of Minnesota, USA, which is one of the most important datasets for evaluating recommendation algorithm. Movielens dataset contains about 100,000 ratings obtained from 943 users for 1,682 movies, and each user has rated at least 20 movies. Movies are rated on an integer scale of 1 to 5 [29]. (2) Netflix dataset comes from Netflix's movie rental website. Netflix published the dataset in 2005 and set up a netflix prize to solicit recommendation algorithms and architectures which can increase the performance of RS by 10%. This dataset contains about 1 billion ratings of 17,770 movies from 480,189 anonymous users [30]. (3) Bookcrossing dataset is crawled by Ziegler from the bookcrossing community. It contains 1,149,780 ratings obtained from 278,858 users for 271,379 books. The dataset contains simple demographic information (age, location, book title, book publishing era, publishing house, etc) of users. Ratings are provided on a scale from 1 to 10 [76].

2) THE DATASETS WITH TRUST RELATIONSHIPS

In these datasets, these users express their opinions about items using ratings and trust relationships with other users. The values of the trust relationships are 0 or 1, where 0 represents lack of trust relationship and 1 represents there is a trust relationship between users [33]. (1) Epinions dataset contains 598,329 ratings obtained from 49,289 users for 139,738 different items, and includes 25 categories and 240 subcategories. (2) Tencent dataset is sampled from 50 days of behavioral data of about 200 million registered users, including about 2 million active users, 6,000 items, and 300 million records of historical activity, as well as social networks, user tags, item categories, and item keywords. Table 2 shows the basic statistic of real life datasets. (3) Flixster dataset is a social movie website in which the users can build friendships and rate movies. The rating values of the items are 10 discrete numbers in range [0.5, 5] with step 0.5. The original dataset is very large and the dataset can be tailored according to actual needs [32], [34].

TABLE 2. **The basic statistic of real life datasets.**

Datasets	Users	Items	Ratings	Sparsity(%)	Trust
Movielens	943	1682	100000	93.7	No
Netflix	480189	17770	1,000,000,000	88.3	No
Bookcrossing	278858	271379	1149780	99.9	No
Epinions	49289	139738	598329	99.9	Yes
Tencent	2,000,000	6,000	300,000,000	97.5	Yes

F. TECHNICAL CHALLENGES ON EXISTING METHODS

As the data volume increases, the data types become more and more rich, the application environment becomes more and more complicated, and the existing algorithms mainly face the following major problems [3], [5], [34], [45].

(1) Data sparsity. There are a lot of unknown ratings in user-item matrix, and the sparsity is often more than 99%. Excessive sparsity gives rise to the number of common ratings between objects too few or none, and there are a big deviation in the similarity calculation, which in turn affect the quality of recommendation. Hence, an effective recommendation algorithm must take the data sparsity into account.

(2) Cold start. When a new user or a new item enters the system, there is usually no histories information of the user or lack of users' ratings for the item, so the user cannot be provided with the recommendation service or the item is difficult to be recommended by the system. The usual solutions of this problem are based on using hybrid recommendation techniques combining ratings and content information (such as users' age, users' trust relations, item tags).

(3) Scalability. In online social networks, on the one hand, the amount of data is growing geometrically, on the other hand, it is necessary to recommend useful results for users in time. Therefore, it is essential to consider the issue of computational cost. In this case, the model-based methods are employed to train model parameters offline to improve the efficiency of online prediction, such as user modeling, similarity calculating, and features extracting.

(4) Diversity. For RS, only recommending popular and highly rated items to the active user often results in better recommendation results. However, the user can also easily obtain such item information from other sources, that is, the actual value of such recommendation is not high. Therefore, a good RS should be able to discover items that are difficult to be found by users spontaneously, but meanwhile which also fit the users' interests.

(5) Interpretability. Interpretability is one of the few concerns of current CF-based algorithms. The quality of the algorithms can't be judged based solely on evaluation such as MAE or RMSE. Recommending items to users relying solely on accuracy not only wastes resources but also bring little benefit. If they can't explain the recommended results well, then they can't determine whether the recommended items meet the needs of users, resulting in reducing system reliability. If RS can provide some explanation information when generating recommendations, the reliability of the recommended results may greatly be improved. Meanwhile, they will greatly arouse the users' attention.

III. HYBRID RECOMMENDER SYSTEMS

In order to improve recommendation accuracy and user's satisfaction, and solve the problems of scalability, cold start, and data sparse, many traditional technologies are combined with each other, such as the time context, and trust relationship between users are integrated into RS. For instance, a framework of trust-based RS is shown in Fig. 7.

A. OVERVIEW OF HYBRID RECOMMENDER SYSTEMS

In recent years, with the advent of online social network, recommendation algorithms based on the social network have emerged and attracted more and more people to study. These algorithms make recommendations for an active user based on the ratings of the users that have direct or indirect social relationships with the active user [70]. These methods

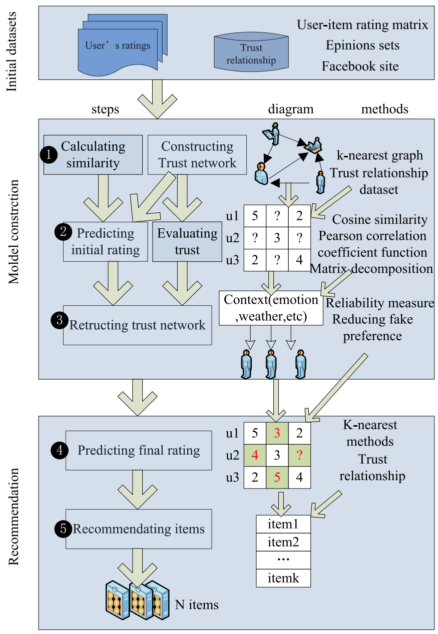


FIGURE 7. A framework of trust-aware RS.

effectively reduce the problems of cold start users using social relationship, and thus improve the accuracy of the recommendation. Some CF-based recommendation methods that fuse trust relationships between users are proposed, such as reliability-based trust-aware CF (RTCF) [32], context-aware social recommendation via individual trust (CSIT) [74], trust-aware RS method based on confidence and Pareto dominance (CPD) [43]. Time decay factor, neighbor relationships are used to enhance similarity measure [25], [73]. In addition, these MF-based methods like SVD, NMF, and PMF are integrated with context relationships, and many improved algorithms are proposed in recent years [47], [48], [65], [75], [78].

B. EMERGING TECHNOLOGIES

In this section, we will introduce some recent approaches and techniques in hybrid CF-based RS.

1) ENHANCED SIMILARITY MEASURES

When the number of common rated items is too small, the similarity is likely to be overestimated using cosine similarity and PCC measures [17]. Some similarity measure based on structural similarity [24] and time decay [25] are proposed to alleviate the problem.

(1) The Similarity Measure Based on Structure. Experimental results demonstrate that the more neighbors who have rated an items, the more accuracy the prediction based on the choice of those neighbors [62]. Therefore, the number of common ratings needs to be considered, that is to say, on the basis of adjusted the cosine similarity, Salton factor of structural similarity is introduced into the similarity measure. The Salton factor can be described as follows [25] (see Eq.32-33):

$$\mathrm{fs}\,(\mathrm{u},\mathrm{v}) = \frac{|\mathrm{I_{uv}}|}{|\mathrm{I_u}| + |\mathrm{I_v}|}, \tag{32}$$

$$\mathrm{fs}\,(\mathrm{i},\mathrm{j}) = \frac{|\mathrm{U_{ij}}|}{|\mathrm{U_i}| + |\mathrm{U_j}|} \tag{33}$$

where fs(u, v) and fs(i, j) denote the Salton factors based on users and items, respectively. The meanings of symbols $\mathrm{I_u}$, $\mathrm{I_v}$, $\mathrm{I_{uv}}$, $\mathrm{U_i}$, $\mathrm{U_j}$ and $\mathrm{U_{ij}}$ are shown in Section 2.3.1.

(2) The Similarity Measure Based on Time Decay. The user's interest changes over time. The fact that u and v have different times rated for the same item means that their interest changes are not synchronized. Therefore, the time decay factor needs to be introduced to weight the similarity between u and v, so that reduce the similarity between users who are far apart in rating time. Likewise, for the similarity of items, the longer the difference between the time that the items i and j rated by u is, the smaller the similarity between i and j is. Therefore, time decay factors for users and items based on the similarity are described as follows [25], [26] (see Eq. 34-35):

$$\mathrm{ft}(\mathrm{u},\mathrm{v}) = \frac{1}{1+\exp(\lambda\,|\mathrm{t_{ui}} - \mathrm{t_{vi}}|)} \tag{34}$$

$$\mathrm{ft}(\mathrm{i},\mathrm{j}) = \frac{1}{1+\exp(\varphi\,|\mathrm{t_{ui}} - \mathrm{t_{uj}}|)} \tag{35}$$

where λ and φ denote the parameters of time decay for users and items, respectively, and $\mathrm{t_{ui}}$ and $\mathrm{t_{uj}}$ denote the time of the items i and j rated by the user u, respectively. $\mathrm{t_{vi}}$ denotes the time of the item i rated by the user v.

2) MEMORY-BASED TRUST-AWARE COLLABORATIVE FILTERING

Trust relationships between users have been introduced into RS as an effective approach to overcome the problems of data sparsity and cold-start [31], [43]. The hybrid approach builds an active user's trust network using trust statements between the users to improve the accuracy of similarities between users. One of the core roles of the trust network is to resolve the neighbor selection between a user's trust statements and its similarity values.

(1) Construct the Trust Network. A trust network for the active user is established based on the Pearson correlation coefficient (PCC) measure and the trust statements as final similarity values. The trust network can be expressed as a directed and weighed graph, in which each node represents a user and an edge represents the trust statement between two users. The trust relationships between two users can be calculated as follows [31] (see Eq. 36):

$$\mathrm{T_{uv}} = \frac{\mathrm{d_{max}} - \mathrm{d_{uv}} + 1}{\mathrm{d_{max}}} \tag{36}$$

where $\mathrm{T_{uv}}$ denotes the trust statement between the users u and v, $\mathrm{d_{max}}$ represents the maximum propagation distance which can be set to any positive integer value (e.g., 4), and $\mathrm{d_{uv}}$ indicates the distance between the users u and v. Fig. 8 is

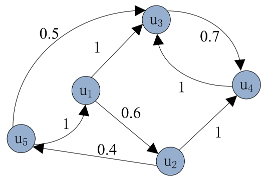


FIGURE 8. An example of trust network

an example of trust network. As shown in Fig. 8, nodes and edges represent users and trust statements between users, respectively. The user u_1 has a trust statement in the user u_2 with the value 0.6, and a trust statement in the user u_3 with the value 1. The user u_5 has a trust statement in the user u_1 with the value 1.

However, the explicit trust relationship between users may not exist in some datasets, in this case, the trust statement can be calculated according to the user-item rating matrix, and the type of the trust statement is called as implicit trust statement, which can be calculated as follows [18], [62] (see Eq. 37):

$$T_{uv} = \frac{|A_{uv}|}{|A_u|} \tag{37}$$

where A_u denotes the set of items rated by the user u, and A_{uv} denotes the set of common rated items by u and v.

(2) Adjust Similarity Measure between Users. Usually, the user-item rating matrix is very sparse so that it may be useful to combine the rating matrix with the trust network to reduce data sparsity [31]. In [32] and [33], according to combing ratings with trust relationships between users, the adjusted weight w_{uv} between users u and v can be described as follows [32] (see Eq. 38):

$$w_{uv} = \begin{cases} \dfrac{2 * sim_{uv} * T_{uv}}{sim_{uv} + T_{uv}}, & sim_{uv} + T_{uv} \neq 0, \text{ and} \\ & sim_{uv} * T_{uv} \neq 0 \\ sim_{uv} & sim_{uv} \neq 0, \text{ and } T_{uv} \\ T_{uv}, & sim_{uv} = 0, \text{ and } T_{uv} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

where sim_{uv} denotes the similarity between the users u and v, which is calculated as Eq.(2).

In [62], another similarity measure combining trust network with use-based similarity by PCC is proposed as follows (see Eq.39):

$$w_{uv} = \alpha \cdot sim_{uv} + (1-\alpha) \cdot T_{uv} \tag{39}$$

(3) Predict Initial Ratings. By employing Eq.(3), the initial ratings of unknown items for the active user u on item i is calculated as follows (see Eq. 40):

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in N_u} w_{uv}(r_{ui} - \bar{r}_v)}{\sum_{v \in N_u} w_{uv}} \tag{40}$$

where N_u denotes a set of neighbors for the user v who has rated the item i, and w_{uv} denotes the adjusted similarity weight between the users u and v.

(4) Measure the Reliability of Ratings. A reliability measure that is suitable for use in any RS based on CF is proposed in [55], which is defined as follows [32], [55] (see Eq.41):

$$R_{ui} = (f_P(P_{ui}) \cdot f_N(N_{ui})^{f_P(P_{ui})})^{\frac{1}{1+f_P(P_{ui})}} \tag{41}$$

where R_{ui} denotes the reliability of a prediction $\hat{r}_{ui}$. P_{ui} and N_{ui} represent the positive and negative factors of the reliability measures, respectively. Accordingly, $f_P(P_{ui})$ and $f_N(N_{ui})$ denote the reliability measure functions of the above positive and negative factors, respectively. Their functions are described as follows [32], [55] (see Eq.42-43):

$$f_P(P_{ui}) = 1 - \frac{\bar{m}}{\bar{m} + P_{ui}} \tag{42}$$

$$f_N(N_{ui}) = (\frac{\max - \min - N_{ui}}{\max - \min})^{\gamma} \tag{43}$$

where P_{ui} and N_{ui} and γ are defined as follows (see Eq.44-46):

$$P_{ui} = \sum\nolimits_{N_u} sim_{uv} \tag{44}$$

$$N_{ui} = \frac{\sum_{v \in N_u} sim_{uv} \cdot (r_{vi} - \bar{r}_v - \hat{r}_{ui} + \bar{r}_u)^2}{\sum_{v \in N_u} sim_{uv}} \tag{45}$$

$$\gamma = \frac{\ln 0.5}{\ln \frac{\max - \min - \bar{v}}{\max - \min}} \tag{46}$$

In general, the larger the value of $f_P(P_{ui})$ is, the more reliable the prediction is. The smaller the value of $f_N(N_{ui})$ is, the more reliable the prediction is.

(5) Reconstruct the Trust Network. According to the above the reliability measurement of the ratings, if the reliability value R_{ui} on the item from the active user u is less than a given threshold value δ, the trust network for the active user u will be rebuilt according to the above trust network re-establishment method, through removing some useless users from the trust network [32].

(6) Predict the Final Ratings and Make a Recommendation. According to Eq.(31), the final ratings for the active user on all the items will be predicted, and the items sorted from big to small will be recommended to the active user u.

3) SOCIAL NETWORKS-BASED MATRIX FACTORIZATION

Furthermore, trust is also adopted in model-based approaches employing MF techniques [70], [74], [81], [82], [85]–[87], [89], [92]–[94]. In [70] and [74], a linear combination of basic MF and a social network-based algorithm is proposed as follows (see Eq.47):

$$R_{ui}^* = \alpha U_u^T V_i + (1-\alpha) \sum\nolimits_{v \in N_u} T_{uv} U_u^T V_i \tag{47}$$

where α denotes used to control the effect of neighbors on the estimated rating.

According to [70], [74], and [81], to optimize the prediction solution in both user latent feature space and user-item rating space, a social MF-based method (SocialMF) using user's trust relationships among users is proposed as follows (see Eq.48):

$$\begin{aligned} L(R,T,U,V) &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i}-g(U_u V_i^T))^2 \\ &+ \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u U_u^T + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i V_i^T \\ &+ \frac{\lambda_T}{2}\sum_{u=1}^{N}((U_u - \sum_{v\in N_u} T_{uv}U_v)^T \\ &\times (U_u - \sum_{v\in N_u} T_{uv}U_v)) \end{aligned} \tag{48}$$

where $g(x)$ indicates the logistic function $g(x) = \frac{1}{1+e^{-x}}$, which bounds the range of $U_u^T V_i$ from 0 to 1. T_{uv} represents the extent of trust between user U_u and user U_v, whichis a positive value $T_{uv} \in [0,1]$. λ_U, λ_V, and λ_T are regularization terms, respectively, and $\lambda_U = \frac{\sigma_R^2}{\sigma_U^2}$, $\lambda_V = \frac{\sigma_R^2}{\sigma_V^2}$, and $\lambda_T = \frac{\sigma_R^2}{\sigma_T^2}$. The graphical model of SocialMF is as shown in Fig.9.

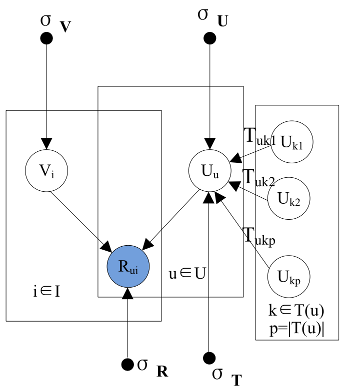


FIGURE 9. The graphical model of the model presented in [81].

Optimize the objective function by conducting gradient descent on $U_u$and V_i as follows (see Eq.49-50).

$$\begin{aligned} \frac{\partial L}{\partial U_u} &= \sum_{i=1}^{M} I_{u,i}^{R} V_i g'\left(U_u V_i^T\right)\left(g(U_u V_i^T) - R_{u,i}\right) \\ &+ \lambda_U U_u + \lambda_T (U_u - \sum_{v\in N_u} T_{u,v}U_u) \\ &- \lambda_T \sum_{\{v|u\in N_v\}} T_{v,u}(\sum_{w\in N_v} T_{v,w}U_w) \end{aligned} \tag{49}$$

$$\begin{aligned} \frac{\partial L}{\partial U_u} &= \sum_{u=1}^{N} I_{u,i}^{R} U_v g'\left(U_u V_i^T\right)\left(g(U_u V_i^T) - R_{u,i}\right) \\ &+ \lambda_V V_i \end{aligned} \tag{50}$$

where $g'(x)$ indicates the derivative of logistic function, i.e., $g'(x) = \frac{e^{-x}}{(1+e^{-x})^2}$.

According to [70], [82], and [88], the concept of social trust circles from available rating data combined with social network data is proposed, and some social factors: user personal interest, interpersonal interest similarity, and interpersonal influence are incorporated into the MF model, and the proposed personalized recommendation model (PRM) is described as follows (see Eq.51):

$$\begin{aligned} &L(R,U,V,P,S,W,Q) \\ &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i}-g(U_u V_i^T))^2 \\ &+ \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u U_u^T + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i V_i^T \\ &+ \frac{\lambda_S}{2}\sum_{u=1}^{N}((U_u - \sum_{v} S_{uv}U_v)^T \\ &\times (U_u - \sum_{v} S_{uv}U_v)) \\ &+ \frac{\lambda_W}{2}\sum_{u=1}^{N}((U_u - \sum_{v} W_{uv}U_v)^T \\ &\times (U_u - \sum_{v} W_{uv}U_v)) \\ &+ \frac{\lambda_Q}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} |H_u|(Q_{u,i}-g(U_u V_i^T))^2 \end{aligned} \tag{51}$$

where S_{uv} and W_{uv} are the normalized interpersonal interest similarity matrix, and interpersonal influence similarity matrix, respectively. H_u indicates the normalized number of items that user u has rated. λ_S, λ_W, and λ_Q are regularization terms.

According to [85], the average-based regularization and individual-based regularization methods are introduced into the MF framework to improve RS (see Eq.52-53):

$$\begin{aligned} &L_1(R,U,V) \\ &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i}-g(U_u V_i^T))^2 \\ &+ \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u U_u^T + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i V_i^T \\ &+ \frac{\lambda_A}{2}\sum_{u=1}^{N} \left|\left|U_u - \frac{\sum_{f\in F^+(u)} sim(u,f)\times U_f}{\sum_{f\in F^+(u)} sim(u,f)}\right|\right|_F^2 \end{aligned} \tag{52}$$

$$\begin{aligned} &L_2(R,U,V) \\ &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i}-g(U_u V_i^T))^2 \\ &+ \frac{\lambda_U}{2}\sum_{u=1}^{N} U_u U_u^T + \frac{\lambda_V}{2}\sum_{i=1}^{M} V_i V_i^T \\ &+ \frac{\lambda_I}{2}\sum_{u=1}^{N} sim(u,f)||U_u - U_f||_F^2 \end{aligned} \tag{53}$$

where sim(u,f) indicates the similarity function. The smaller the value of sim(u, f) is, the greater the distance between U_u and U_f is. Otherwise, the larger the value of sim(u, f) is, the smaller the distance between U_u and U_f is. These values of the eigenvectors U_u and U_f are solved by using the similarity of users u_u and u_f.

According to [85]–[88], user social status, homophily theory, and social tags are fused into MF model to improve RS.

$$\begin{aligned} &L(R,G,U,V) \\ &= \sum_{U,H}(G-UHU)^2 \\ &+ \frac{\lambda_1}{2}\max\sum_{i}^{N}\sum_{f=u+1}^{N}\left\{0, f(r_i - r_j)(U_i H U_j^T - U_i H U_j^T)\right\} \\ &+ \frac{\lambda_2}{2}\sum_{i=1}^{N}\sum_{j=i}^{N}\varphi(u,f)||U_u - U_f||_F^2 \end{aligned} \tag{54}$$

where r_i and r_j denote the level of social status from users u_i and u_j. The matrix U represents users' preference matrix, each row represents the user, and each column represents the user's preference. H represents the degree of association between users' preferences. Among them, $f(r_i - r_j) = \sqrt{\frac{1}{1+\log(r_j)} - \frac{1}{1+\log(r_i)}}$, φ(i,j) denotes the homogeneous coefficient between users u_i and u_j, and $\varphi(i,j) = \alpha \frac{\sum_{k=1}^{m} rate_{ik} \times rate_{jk}}{\sqrt{\sum_{k=1}^{m} rate_{ik} \times \sum_{k=1}^{m} rate_{jk}}} + (1-\alpha)\frac{|N_i \cap N_j|}{|N_i \cup N_j|}$. Here, $rate_{ik}$ indicates the rating on $item_k$ from user u_i, N_i and N_j denote the numbers of users trusted by user u_i and u_j.

According to [34], [83], [87], and [89], user attributes and item labels are integrated into MF model to reduce data sparsity. For instance, Ji and Shen [34] propose a MF-based model fusing user interest weight and item relevance weight as follows (see Eq.55):

$$\begin{aligned} L(R,P,Q,S) &= \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{u,i}^{R}(R_{u,i} - P_u S_{ui} Q_i^T)^2 \\ &+ \frac{\lambda_U}{2}\sum_{u=1}^{N} P_u P_u^T + \frac{\lambda_V}{2}\sum_{i=1}^{M} Q_i Q_i^T \\ &+ \frac{\lambda_Q}{2}\sum_{u=1}^{N}\sum_{f=F^+(u)} sim(u,f)|\,|P_u S_u - P_f S_f|\,|_F^2 \end{aligned} \tag{55}$$

where P_u represents interest weight on all tags from user u, and Q_v represents relevance weight on all keys. S_{ui} denotes the degree of relevance between user u and item i. Both S_u and S_f denote the similarities between user tags and item keys.

In [65], a novel hybrid MF-based RS model (Hybrid Matrix Factorization, HMF) is proposed, which employs hypergraph theory to express the interior relationship of social network, including user's feature, item's feature, and contextual information are all integrated to MF model. The proposed model is described as follows (see Eq.56):

$$\begin{aligned} L&\left(R^{T_x}, P^{T_x}, Q^{T_x}, SC^{T_x}, SU^{T_x}, SS^{T_x}\right) \\ &= \frac{1}{2}\sum_{(u_i,s_j)\in T_x}\left(R_{ij}^{T_x} - \widehat{R}_{ij}^{T_x}\right)^2 \\ &+ \frac{\lambda}{2}\left(\left\|P^{T_x}\right\|_F^2 + \left\|Q^{T_x}\right\|_F^2\right) \\ &+ \frac{\alpha}{2}\sum_{T_x}((P_i^{T_x} - \sum_{u_v \epsilon N_{u_i}^{T_x}} SC_{iv}^{T_x} P_v^{T_x}) \\ &\times (P_i^{T_x} - \sum_{u_v \epsilon N_{u_i}^{T_x}} SC_{iv}^{T_x} P_v^{T_x})^T) \\ &+ \frac{\beta}{2}\sum_{T_x}((P_i^{T_x} - \sum_{u_v \epsilon N_{u_i}^{T_x}} SU_{iv}^{T_x} P_v^{T_x}) \\ &\times (P_i^{T_x} - \sum_{u_v \epsilon N_{u_i}^{T_x}} SU_{iv}^{T_x} P_v^{T_x})^T) \\ &+ \frac{\gamma}{2}\sum_{T_x}((P_i^{T_x} - \sum_{s_v \epsilon M_{s_j}^{T_x}} SS_{jv}^{T_x} P_v^{T_x}) \\ &\times (P_i^{T_x} - \sum_{s_v \epsilon M_{s_j}^{T_x}} SS_{jv}^{T_x} P_v^{T_x})^T) \end{aligned} \tag{56}$$

where T_x indicates the first x cluster in training dataset. $R_{ij}^{T_x}$ and $\widehat{R}_{ij}^{T_x}$ denote the real and predicted ratings on item s_j from user u_i, respectively. The parameter α controls the factor of rating similarity between users, and β and γ control the factor of similarities between user features and between item features, respectively.

By introducing some social factors such as trust relationships between users, and user's social status in social networks into the matrix factorization model, the problems of data sparsity and cold start can be alleviated to some extent. For example, Fig.10 shows a decomposition and recommendation process for the user-item rating matrix based on the user trust relationship. The user's trust relationship graph consists of 5 nodes and 10 edges, where the node represents the user and the edge represents the trust relationship between two users. The extent of trust between users is represented by the value of the range of [0, 1]. Fig. 10(a) reports the results of predicted ratings based on PMF method, but we can't predict user u_4's preference for any item because we can't obtain the user u_4's neighbor relationship through the user-item rating information. Fig. 10(b) reports the results of predicted ratings based on social matrix factorization method, and we can predict user u_4's preferences through the trust relationships between users.

4) REDUCE DIMENSIONALITY

To solve the high dimensionality in RS, some dimensionality reduction techniques are used to find the most similar items and users in each cluster of items and users which can significantly improve the scalability of the recommendation method [78]. Clustering and SVD are usually used techniques in RS.

(1) Singular Value Decomposition. In RS, SVD is used for dimensionality reduction, and it can also be used directly for prediction tasks. The prediction process is as follows [27], [78]:

Step 1: Covert the rating matrix to the new dense matrix D. The user-item rating matrix $R_{m\times n}$ is mapped to the dense matrix $D_{m\times n}$ using SVD techniques as Eq.(18), i.e., for finding the new coordinates of users and items in the matrix $D_{m\times n}$, we convert raw data to the k-dimensions space as follows (see Eq.57-58):

$$U_{Trans} = R_{m\times n} \times V_{n\times k} \times \Sigma_{k\times k}^{-1} \tag{57}$$

$$V_{Trans} = R_{m\times n} \times U_{n\times k} \times \Sigma_{k\times k}^{-1} \tag{58}$$

where U_{Trans} and V_{Trans} are new coordinates of users and items in the k dimensions space.

For instance, the matrix in Fig. 9(a) is denoted as R, which can be decomposed into U, V, and Σ. We can obtain the approximation of R by taking the first 2-dimensional data, i.e.,

$$U' = \begin{pmatrix} 0.557 & 0.733 \\ 0.503 & -0.475 \\ 0.439 & 0.020 \\ 0.4356 & -0.472 \\ 0.233 & 0.118 \end{pmatrix}, \quad V' = \begin{pmatrix} 0.695 & 0.095 \\ 0.266 & 0.345 \\ 0.344 & 0.559 \\ 0.239 & -0.369 \\ 0.435 & -0.646 \\ 0.285 & 0.068 \end{pmatrix},$$

$$\Sigma' = \begin{pmatrix} 11.65 & 0 \\ 0 & 5.767 \end{pmatrix}.$$

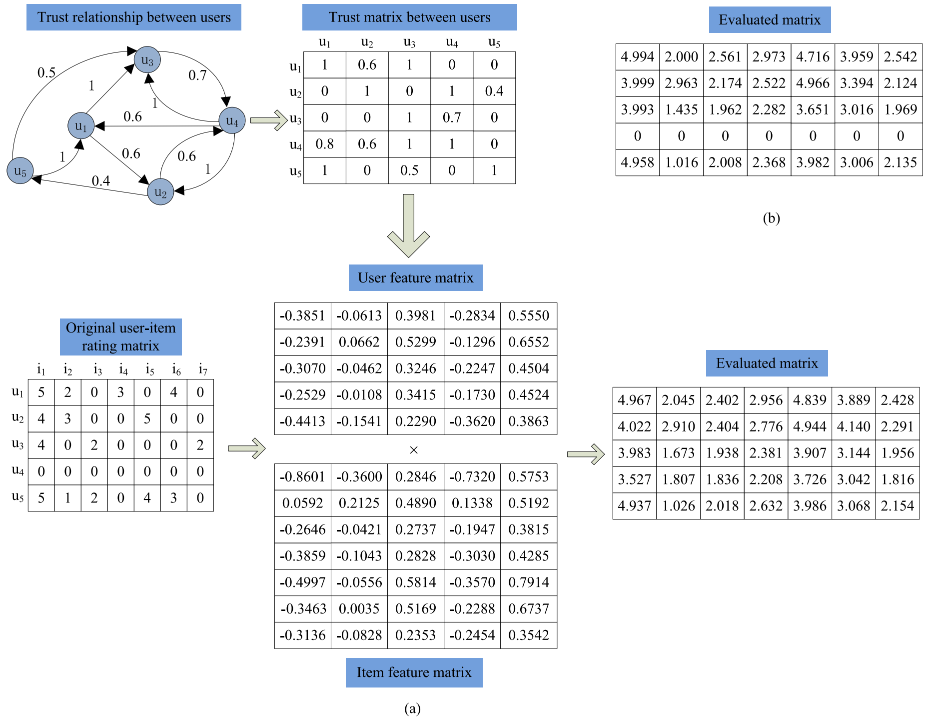


FIGURE 10. An example of social matrix factorization. (a) The predicted rating matrix by the social matrix factorization. (b) The predicted rating matrix by PMF.

U' and V' are projected in 2-dimensional space and plotted in Fig. 9.

When a new user u_{new} who shares the rating as [5, 3, 4, 0, 1, 2] arrives, to obtain the coordinate of the new user in the 2-dimensional space, the following calculation is performed as:

$$u_{new} = [5, 3, 4, 0, 1, 2]^T \times V' \times \Sigma'^{-1} = [0.572, 0.561]$$

As can be seen in Fig. 11, it can be found the user u_1 close to the new user u_{new} for forming k-nearest neighbors.

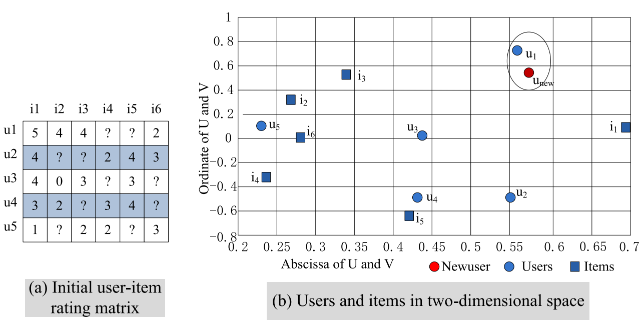


FIGURE 11. Two-dimensional space of applying SVD for users, items and new users. (a) Initial user-item rating matrix. (b) Users and items in two-dimensional space.

Step 2: Normalize the rating matrix D. The matrix D is normalized employing Z-score to the $Z_{m\times n}$ by $Z_{ij}^{(u)} = \frac{U_{ij}-\bar{U}_i}{\pi_i}$ and $Z_{ij}^{(i)} = \frac{I_{ij}-\bar{I}_j}{\sigma_j}$, respectively. Here $\bar{U}$ and π denote the average ratings and standard deviation for users, respectively, and $\bar{I}$ and σ denote the average ratings and standard deviation for items, respectively.

Step 3: Apply SVD method on the matrix Z. i.e., the matrix Z is decomposed using SVD to obtain the new U, S and V.

Step 4: Obtain an approximation of Z. According to the low-rank matrix U, S and V, we can obtain a new matrix, denoted by $\hat{Z}$.

Step 5: Predict the unknown ratings. We can predict the unknown ratings based on $\hat{r}_{ij}^{(u)} = \bar{U}_i + \pi_i Z_{ij}^{(u)}$ or $\hat{r}_{ij}^{(i)} = \bar{I}_j + \sigma_j Z_{ij}^{(i)}$.

(2) Spectral Clustering. The idea of spectral clustering is derived from the theory of spectral partitioning. Its essence is to convert the clustering problem into the optimal partitioning problem, so as to achieve the goal that the distance between data points inside the subgraph is as similar as possible, and the distance between the subgraphs is as far as possible. The spectral clustering considers the data points as a weighted undirected graph G (V, E), where V is the set of sample points and E is the weighted edges set, whose values are the similarity between the sample points. The process of spectral clustering is to divide the undirected graphs according to the classification criteria so that the similarity within each sub-graph is enough large and the similarity between subgraphs is enough small [24], [25].

CF-based recommendation algorithm based on spectral clustering is described as follows [24], [59]:

Step 1: Construct the weighted undirected graph A. For the user-item rating matrix R, each user corresponds to a vertex in the graph, and the similarity between the two users is calculated using cosine similarity to obtain the user similarity matrix A, $\forall a_{ij} \epsilon A$, $a_{ij} = \text{sim}(u_i, u_j)$. Where a_{ij} is the weight of the associated edge between the nodes v_i and v_j. When there is no edge between two nodes, the associated weight is set to 0, denoted as $a_{ij} = 0$.

Step 2: Obtain the degree matrix D. We add the values of each column of the similarity matrix A, and the result are placed on the diagonal to obtain a diagonal matrix D of $N * N$, namely a degree matrix. The values of diagonal element in D are denoted as $d_{jj} = \sum_j^N a_{jj}$.

Step 3: Obtain the Laplacian matrix. The results of D-A are denoted by the Laplacian matrix, i.e. L = D-A.

Step 4: Normalize the matrix L according to $L = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = D^{-\frac{1}{2}} (D - A) D^{-\frac{1}{2}} = D^{-\frac{1}{2}} D D^{-\frac{1}{2}} - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = E - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, so the normalization of L is transformed into the normalization of A.

Step 5: Find the first k feature values of L. Suppose that $\lambda_1, \lambda_2, \ldots, \lambda_k$ are the first k feature values are denoted as $\lambda_1, \lambda_2, \ldots, \lambda_k$, where $\lambda_i \geq \lambda_j$ and $i<j$. Correspondingly, we can get the eigenvectors $v_1, v_2, \ldots, v_k$.

Step 6: An eigenmatrix of $N*k$ is composed of these eigenvectors. The partitioning of the graph represented by the matrix is performed using k-means algorithm. We can get the classification of N nodes using clustering algorithms. All users are divided into L classes by using the spectral clustering, i.e. $U_1, U_2, \ldots, U_L$ denote the L classes, respectively, among them, $U_i \cap U_j = \emptyset$, $U_1 \cup U_2 \cup \ldots U_L = U$.

Step 7: Calculate the similarity between the elements in each cluster. Obtain the similarity matrix W, which is a symmetric matrix, and the elements on the diagonal are all 1.

Step 8: Calculate the preference of each user u_i on each item j. The preference of u_i on item j is denoted as $\text{pre}_{ij} = \sum_{k \in B \cap K(j)} W_{jk} * R_{ik}$, where K(j) represents the set of the first k similar items of item j. Here B is the collection of items that the user u has already purchased.

Step 9: Recommend items to users. The items in the set S are sorted according to the interest degree of the user u, and the first N items are the final recommendation results. Here, S is the collection of items that have been purchased or liked by the user u.

For instance, Fig. 12 shows the recommendation process using the spectral clustering. There are 7 users and 10 edges in the graph, and the 7 users are divided into two clusters, i.e., $U_1 = \{u_3, u_4, u_5, u_6\}$, and $U_2 = \{u_1, u_2, u_7\}$.

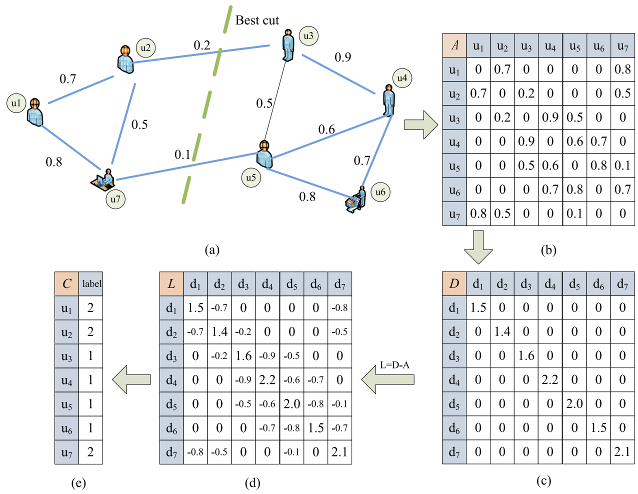


FIGURE 12. The process of spectral clustering. (a) neighbor graph. (b) similarity matrix. (c) diagonal matrix. (d) laplacian matrix. (e) classification.

5) GUASSIAN MIXTURE MODEL

In fact, each user has multiple interests, and thus the user may belong to multiple user groups [48]. According to [35], [48], and [77], the Gaussian mixture model proposed by Hofmann is employed as the basis of clustering, and suppose that the conditional probability of the rating on the item t, which belongs to the group z, obeys the Gaussian distribution (see Eq. 59).

$$p\,(r\,|\,u, t) = \sum\nolimits_{z_k \in Z} p\,(z_k\,|\,u)\, p(r|\mu_i^{z_k}, \Sigma_i^{z_k}) \tag{59}$$

where p(r|u,t) denotes the joint probability of the user u and the item t. Then we use the expectation maximization (EM) algorithm and maximum likelihood estimation to solve the model [77], [78].

(1) Initialize means and variances of the model $\{\mu_k^{(0)}, \Sigma_k^{(0)}, \pi_k^{(0)}\}$.

(2) E step: expectation. Using the estimates of $\theta^{(t)} = \{\mu_k^{(t)}, \Sigma_k^{(t)}, \pi_k^{(t)}\}$ to calculate the estimate of $p\,(z_k\,|\,u, t, v)$ as follows (see Eq. 60):

$$p\,(z_k\,|\,u, t, v) = \frac{p(z_k|u)p\left(r|\mu_a^{z_k}, \Sigma_a^{z_k}\right)}{\sum_{z_k \epsilon Z} p\left(r|\mu_{t,z}, \Sigma_{t,z}\right) p(z_k|u)} \tag{60}$$

(3) M step: Maximization. Using the estimates of $p\,(z_k\,|\,u, t, v)$ update the estimates of the model parameters as follows [77] (see Eq. 61-63):

$$p\,(z_k\,|\,u) = \frac{\sum_{\left\{u',t,r\right\}:u'=u} p(z_k|u, t, r)}{\sum_{z_k \epsilon Z} \sum_{\left\{u',t,r\right\}:u' \neq u} p(z'|u, t, r)} \tag{61}$$

$$\mu_{t,z} = \frac{\sum_{\left\{r',t,r\right\}:t'=t} rp(z_k|u, t, r)}{\sum_{\left\{u',a,r\right\}:t'=t} p(z_k|u, t, r)} \tag{62}$$

$$\Sigma_{t,z}^2 = \frac{\sum_{\left\{u',t,r\right\}:t'=t} \left(r - \mu_{t,z}\right)^2 p(z|u, t, r)}{\sum_{\left\{u',t,r\right\}:t'=t} p(z|u, t, r)} \tag{63}$$

(4) Check For Convergence. Execute E and M steps alternately, until the error of the parameters is converged, and the model parameters are been obtained.

(5) Predict the Ratings. The ratings of the unknown items by the user is predicted as follows [48], [77] (see Eq. 64):

$$r_{ui}^{(plsa)} = E\,(p\,(r\,|\,u, t)) = \sum\nolimits_{z \epsilon Z} p(z|u)\mu_{t,z} \tag{64}$$

In addition to the users' multi-interest features, a hybrid algorithm makes up for a single recommendation based on the user model by analyzing item similarity. The recommendation method is described as follows [48] (see Eq. 65):

$$r_{ui} = \lambda r_{ui}^{(plsa)} + (1-\lambda) r_{ui}^{(knn)} \tag{65}$$

where $r_{ui}^{(knn)}$ denotes that the predicted ratings is calculated using the item-based similarity as Eq.(6).

For top-N recommendation task, some novel CF-based recommendation approaches have been proposed to improve the recommendation performance, especially in the presence of sparse data and cold start [107]–[109], [112], [113], [122]. For example, according to [108], a bicluster neighborhood-based CF algorithm is proposed, and the ranking rating of a candidate item i from user u is calculated as follows (see Eq. 66):

$$r\left(u,i^{'}\right) = global(u,i^{'}) \times local(u,i^{'}) \tag{66}$$

where $global(u,i^{'})$ and $local(u,i^{'})$ denote the average global and local distances between user u and item $i^{'}$ based on bicluster similarity, respectively.

According to [125] and [127], based on the idea that the more the user acts on an item, the higher the confidence level of the corresponding preference is, a concept of confidence in the sample instance is proposed as follows (see Eq. 67):

$$c_{ui} = 1 + \alpha r_{ui} \tag{67}$$

where r_{ui} denotes the frequency of user behavior, and α is the control coefficient. The objective function of weighted LFM fusing the confidence level is as follows [127] (see Eq.68):

$$\min \sum\nolimits_{u,i} c_{ui}(p_{ui} - w_u^T h_i)^2 + \lambda(\sum\nolimits_u ||w_u||^2 + \sum\nolimits_i ||h_i||^2 \tag{68}$$

where p_{ui} is a binary value, 0 means a negative sample, and 1 means a positive sample. w_u and h_i represent the characteristic factors of the user and the item, respectively.

According to [106] and [128], based on the implicit feedback, a CF model fusing the social interactions and the influence between users is proposed as follows (see Eq.69):

$$F_{i,j} = U_i V_j + \sum\nolimits_{k \in N_i} \frac{\omega_{ik}}{|N_i|} U_k V_j \tag{69}$$

where N_i denotes the set of friends of active user u_i, and ω_{ik} indicates a weight parameter, which reflects the extent that the friend u_k affects the active user u_i.

C. EXAMPLES

With the rapid increase in the volume of data, data sparsity and high dimensionality have become urgent problems to be solved in RS. Therefore, in recent years, more and more studies focus on solving the problems of data sparsity and high dimensionality in RS. In this section, we present a list of references on hybrid CF-based recommendation algorithms in recent years as Table 3. For instance, Zahra *et al.* [29] propose a k-means clustering-based recommendation approach to solve the scalability issues related to conventional RS. Moradi and Ahmadian [32], Azadjalal *et al.* [33], and Xia *et al.* [55] propose a reliability-based trust-aware CF appraoch to promote the precision of the trust-based CF. At first, the proposed method construct a initial trust network according to similarity and trust relationship between users, and then evaluate the reliability of predictions, finally, the trust network is reconstructed and the final ratings of the missing ratings are predicted [32]. Huang *et al.* [47] and Wang*et al.* [79] propose a CF algorithm based on joint NMF by mining the hidden complex relationships between items to recommend items for users more accurately, which combines the user-based CF with the item-based CF. To solve data sparsity and high dimensionality, Koohi and Kiani [51] and Ramezani *et al.* [72] propose a subspace clustering approach to find neighbor users, and a new similarity method is proposed to calculate the similarity value. Zheng *et al.* [65] propose a novel hybrid recommendation model based on MF approach (Hybrid Matrix Factorization, HMF) by using hypergraph theory to describe contextual information, including user features, item features, and similarities of ratings from users. Pan *et al.* [54] propose a social recommendation approach based on implicit similarity in trust (SocialIT) to exactly reflect social relationships among users. Guo *et al.* [91] propose a novel social recommendation algorithm, which integrates item relationships according to a PMF framework from items' perspective. Ma *et al.* [92] propose a PMF-based factor analysis method to solve the problems of data sparsity and poor prediction accuracy by using both user's social network information and rating records. Yu *et al.* [93] propose a novel recommendation approach by incorporating users' social status into MF model. Li *et al.* [94] introduce social status and bias into the construction of social networks, and propose a social recommendation method-based trust relationship. Li *et al.* [30] propose a MF framework that contains two efficient models, that is, dynamic single-element-based Tikhonov graph regularization NMF (DSTNMF) and dynamic single-element-based CF-integrating manifold regularization (DSMMF), and these models incorporate the graph regularization to address the data sparsity.

IV. DISCUSSION

With the arrival of the era of big data, CF has become one of the most successfully and widely used recommendation approaches, aiming at helping people reduce the amount of time they spend to find out the items they are interested in [1], [2], and [30]. Many existing methods and techniques such as MF, NMF, and SVD are proposed to solve the scalability in RS. With the development of Internet technology and the advent of pervasive computing, data grows geometrically and the problems of data sparsity and high dimensionality have become urgent problems to solve. For this reason, many hybrid CF recommender systems have emerged in recent years. These hybrid recommender systems combine model-based and memory-based techniques with

TABLE 3. Summary of articles on hybrid CF approaches.

Category	Methods	Hybrid models	Advantages	Problems	Metric	Datasets
Neighborhood-based CF	[51]neighborhood-based, constructing item subspace	Subspace clustering	Solve data sparsity	The process of constructing item subspace is complex,	Accuracy, Precision, Recall	ML100k, ML1M, Jester
	[72]UTAOS(Users' Tree Accessed on Subspaces)	Subspace clustering, neighborhood-based	Solve data sparsity, reduce the dimensions	The process of constructing user subspace is complex, and the interpretability of prediction is poor	MAE	Movielens, Jester
	[32]RTCF	Trust network, trust-based reliability measure	The accuracy and the reliability of the predictions are improved	The interpretability of reliability measure is poor	MAE, MAUE,RC,UC	Epinions, Flixster
Dimensionality reduction	[48]Gaussian pLSA-item-based CF hybrid model	pLSA, GM, item-based CF	Improve accuracy of recommendation	It does not consider context information, when user-item rating matrix is too sparse, the performance of recommendation will reduce	MAE	Movielens
	[29]k-means clustering	k-means, Fuzzy C-means clustering, EM	Solve scalability, centroid selection	The time complexity of the calculation is higher	MAE	Movielens, FilmTrust, Bookcrossing
Hybrid recommendation model combining several algorithms	[30] DSMMF and DSTNMF	DSMMF and DSTNMF, user and item content information	Improve the high prediction accuracy while data sparsity, the proposed algorithm overcome the dimensionality curse and has certain practicability.	The framework of the RS is too complicated and the computational complexity of user-graph-based DSTUNMF algorithm is high	MAE, RMSE	Movielens, Epinions
	[47][79]Joint Nonnegative Matrix Factorization (JNMF)	User-based and item-based collaborative filtering	Good explanation, solve data sparsity and improve accuracy of the predictions	The derivation of the training model is complex and the time complexity is high	MAE	Movielens
	[35]GGCF	Gaussian-Gamma model, Bayesian CF model	Construct a hierarchical Bayesian model, the model is more robust	It does not consider context information, so it solve data sparsity	MAE, RMSE	Movielens
	[75]Regularized single-element-based NMF	NMF-based CF model with a single-element-based approach	Improve high accuracy and reduce computational complexity	When the data sparsity is too large, the recommendation accuracy will reduce	RMSE,NMSE	Movielens, Jester, Dating Agency
Social matrix factorization model	[64]SPF	Social Poisson factorization, probabilistic matrix factorization	incorporate social network information into the traditional factorization method	The modeling process is complicated,	CRR, NCRR	Epinions, Flixster, Douban, Ciao
	[92] SoRec	Social recommendation using PMF, users' social network information	Solve the data sparsity and poor prediction accuracy problems	Only use inter-user trust information, and ignore the information diffusion or propagation between users	MAE	Epinions
	[70] RSTE	Recommendation with social trust ensemble (RSTE)	Solve the problems of data sparsity and inaccuracy prediction	When both the user-item rating matrix and the trust realtions of a social network are very sparse, the diffusions of trust relations become inevitable	MAE, RMSE	Epinions, EachMovie
	[81] SocialMF	Matrix factorization based model for recommendation in social rating networks	Reduce the problems with cold start users	The cold start users need to be connected to the social network	MAE, RMSE	Flixster, Epinions
	[82] PRM	Personal interest, interpersonal interest similarity, and interpersonal influence are fused into a unified personalized recommendation model	Solve the cold start and sparsity problems	Only consider user historical rating information and interpersonal relationship of social network, and ignore user location information	MAE, RMSE	Yelp, Movielens
	[83] a matrix factorization framework with social regularization	Biclustering, social regularization, incorporate social network information to benefit RS	Improve the accuracy of recommendation	The cold-start problem, the influence from distance friends who are multiple hops away, the time-series information, the place information	Precision, recall	Dilicious
	[74]Context-aware social recommender system via individual trust among users (CSIT)	context-aware enhanced model based on Gaussian mixture model, social network and contexts	The recommendation performance is improved and the data sparsity is alleviated	The establishment of the training model is complicated	MAE, RMSE	Epinions, douban
	[77]Latent semantic models	Expectation Maximization, Clustering, dimension reduction	Higher accuracy	Cannot solve the problem of excessive sparse data	MAE	EachMovie
	[27] [78]SVD-hybrid CF	SVD, hybrid item-based CF, hybrid user-based CF	Sparsity and scalability	It need to fill data before producing recommendation	MAE, precision, recall, f1	Movielens, yahoo
	[65]Hybrid matrix factorization(HMF)	Social network, hypergraph topology, matrix factorization	Cold start problem is tackled and sparse rating is dealt with	The computational complexity is high, and the number of the parameters is too high	MAE, RMSE	Movielens, epinions, douban
Bayesian model	[35] Gaussian-Gamma CF(GGCF)	Gaussian-Gamma distribution, Gibbs samples	Solve the problems of robust and penalty terms of the latent features	The model establishment is relatively complex. When the distribution of the data is skewed, neither GGCF nor regularized CF are suitable.	MAE, RMSE	Movielens, book-crossing
	[70]PMF-Gaussion model	Gaussian model, Probabilistic matrix factorization	The mechanism of trust propagation is integrated into MF model, lead to a substantial increase in recommendation accuracy, in particular for cold start users	Bayesian inference process is complicated.	RMSE	Epinions, Flixster

context relationships such as the trust relationship between users, or integrate multiple recommendation techniques to improve the performance of recommendations. Experimental results indicate that these hybrid RS can enhance the performance of RS.

Although the CF-based recommender system still has some shortcomings, such as sparsity, cold start, and scalability, compared with the content-based filtering methods, CF has the following advantages: 1) It can filter information that is difficult to analyze automatically through machines, such as artwork, music, video, etc. 2) It can share the experience of others, avoiding incomplete and inaccurate content analysis, and can filter some complex and difficult to describe concepts (such as information quality, and personal taste). 3) It has the ability to recommend new information, and find the content that is completely dissimilar in content. The recommended products are usually preferred by users according to the content-based filtering method, and the CF-based filtering method can find the user's potential interests but not yet discovered preferences. 4) It can effectively use feedback information from other similar users to make recommendations. The user's personalized interest preferences can be extracted through less feedback from the user.

In the past, the traditional RS mainly relied on the user-item rating matrix to make recommendations. However, the user-item rating matrix is only one aspect of the user's historical behaviors, and it ignores the user's dynamic process and contextual information for rated items. With the appearance of various algorithms and variants, the accuracy of the recommendation was improved to a certain extent. However, in the face of big data challenges today, it is difficult to make accurate recommendations using only extremely sparse data information, and the recommended results are difficult to satisfy users. In fact, the effect of the recommendation is not only related to the historical behavior data of items from users (such as user-item ratings), but also has a great correlation with the interaction behaviors among users, time, location, mood, etc. Therefore, a good RS should not only mine the user's historical behavior information, but also take into account the user's context information (trust relationships, friend relationships, user tags, item attributes, time information, location, etc) as much as possible. Many studies show that the hybrid algorithm which integrates various social factors has alleviated the problems of data sparsity and cold start to some extent [5], [9], [33], [34], [61], [70], [85], [87], [98].

V. CONCLUSIONS

In the era of big data, RS helps users spend less time finding their favorite items. In the paper, we survey the recent articles on solving data sparsity and high dimensionality, summarize the approaches and techniques of the traditional and hybrid CF-based recommender systems, and discuss the major challenges and the advantages of the CF-based RS.

Some hybrid models are proposed through integrating various latent factor models with various users' social relationships, and the results have indicated that data dimensions are reduced, recommendation accuracy is improved effectively, and scalability of RS is enhanced based on these models [5], [22], [67], [70], [71], [82], [83]. In hybrid recommender systems, the trust is an important concept that recently has attracted lots of attention from academia and industry. Various social factors have been considered in recommendation algorithms and a variety of recommendation models are produced, such as RTCF, SocialMF, PRM, RSTE, and ISRec [32], [50], [52], [54], [61], [65], [70], [71], [82], [85], [88].

Although various influence factors are considered to improve the performance of RS, it will increase the parameter setting and time complexity of the model. In addition, it is difficult to obtain the optimal value due to too many parameters. With the development of deep learning technology, deep learning has gradually been applied in RS due to strong feature representation, and it can learn the latent item association from the user-item rating directly for predictive recommendation without employing a similarity measure [95], [97]–[102]. In recent years, some recommendation models based on deep learning and tensor factorization have been proposed, such as DeRec [99], SADE [101], DRMF [100], DLNN [102], TFCF [114], CoTF [115], and WHBPR [117], and these models exhibit higher recommendation precision compared with state-of-the-art recommendation algorithms [95]–[97], [99], [105], [118], [119], [121]. Therefore, in the future research of RS, to achieve better performance, we should focus on how to use deep learning technology to solve the problems of data sparsity and cold start.

REFERENCES

[1] L.-C. Wang, X.-W. Meng, and Y.-J. Zhang, "Context-aware recommender systems," *J. Softw.*, vol. 23, no. 1, pp. 1–20, 2012.
[2] F. Ricci, L. Rokach, and B. Shapira, "Context-aware recommender systems," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2010, pp. 217–253.
[3] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, 2012.
[4] G. Georg and C. Ehmig, "Recommendations in taste related domains: Collaborative filtering vs. social filtering," in *Proc. Int. ACM SIGGroup Conf. Supporting Group Work*, Sanibel Island, FL, USA, Nov. 2007, pp. 127–136.
[5] Q.-L. Gao, L. Gao, J.-F. Yang, and H. Wang, "A preference elicitation method based on users' cognitive behavior for context-aware recommender system," *Chin. J. Comput.*, vol. 38, no. 9, pp. 1767–1776, 2015.
[6] J. Huang, X.-Q. Cheng, H.-W. Shen, T. Zhou, and X. Jin, "Exploring social influence via posterior effect of word-of-mouth recommendations," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2012, pp. 573–582.
[7] Z. D. Champiri, S. R. Shahamiri, and S. S. B. Salim, "A systematic review of scholar context-aware recommender systems," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1743–1758, 2015.
[8] M. Jugovac and D. Jannach, "Interacting with recommenders—Overview and research directions," *ACM Trans. Interact. Intell. Syst.*, vol. 7, no. 3, pp. 1–46, 2017.
[9] X. W. Meng, S. D. Liu, Y. J. Zhang, and X. Hu, "Research on social recommender systems," *J. Softw.*, vol. 26, no. 6, pp. 1356–1372, 2015.
[10] C. C. Aggarwal, *Recommender Systems*. Cham, Switzerland: Springer, 2016, pp. 268–282.
[11] X. Zhou, J. He, G. Huang, and Y. Zhang, "SVD-based incremental approaches for recommender systems," *J. Comput. Syst. Sci.*, vol. 81, no. 4, pp. 717–733, 2015.
[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[13] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapt. Interact.*, vol. 12, no. 4, pp. 331–370, 2002.
[14] J. Salter and N. Antonopoulos, "CinemaScreen recommender agent: Combining collaborative and content-based filtering," *IEEE Intell. Syst.*, vol. 21, no. 1, pp. 35–41, Jan. 2006.
[15] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artif. Intell. Rev.*, vol. 13, no. 5, pp. 392–408, 1999.
[16] Z. Huang, D. Zeng, and H. Chen, "A comparison of collaborative-filtering recommendation algorithms for e-commerce," *IEEE Intell. Syst.*, vol. 22, no. 5, pp. 68–78, Sep. 2007.
[17] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile Internet applications," *IEEE Access*, vol. 4, pp. 3273–3287, 2016.
[18] Z. Zhang, "Research on personalized recommendation models and algorithm in social networks," Ph.D. dissertation, Dept. Manage. Sci. Eng., Shandong Normal Univ., Jinan, China, 2015.
[19] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, 2015.
[20] B. M. Sarwar, G. Karypis, J. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system—A case study," in *Proc. ACM Web KDD Workshop*, 2000, pp. 1–12.
[21] N. Sherif and G. Zhang, "Collaborative filtering using probabilistic matrix factorization and a Bayesian nonparametric model," in *Proc. IEEE 2nd Int. Conf. Big Data Anal.*, Mar. 2017, pp. 391–396.
[22] D. Rafailidis and F. Crestani, "Learning to rank with trust and distrust in recommender systems," in *Proc. 11th ACM Conf. Recommender Syst. (RecCys)*, 2017, pp. 5–13.

[23] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. T. Reinders. (2011). "Personalised travel recommendation based on location co-occurrence." [Online]. Available: https://arxiv.org/abs/1106.5213
[24] W. Xiao, S. Yao, and S. Wu, "Top-N collaborative filtering recommendation algorithm based on user spectrum clustering," *Comput. Eng. Appl.*, vol. 54, no. 7, pp. 138–143, Apr. 2017.
[25] Q. Li, S. Li, and G. Xu, "Collaborative filtering recommendation algorithm based on spectral clustering and fusion of multiple factors," *Comput. Appl. Res.*, vol. 34, no. 10, pp. 2905–2908, 2017.
[26] Q. Wang, "Research and application of commodity recommendation algorithms based on clustering methods," M.S. thesis, Dept. Electron. Eng., Beijing Jiaotong Univ., Beijing, China, 2014.
[27] J. Lin, X. H. Yan, and B. Huang, "Collaborative filtering recommendation algorithm based on SVD and fuzzy clustering," *Comput. Syst. Appl.*, vol. 25, no. 11, pp. 156–163, 2016.
[28] H. Yu and J. H. Li, "Algorithm to solve the cold-start problem in new item recommendations," *J. Softw.*, vol. 26, no. 6, pp. 1395–1408, 2015.
[29] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems," *Inf. Sci.*, vol. 320, pp. 156–189, Nov. 2015.
[30] Y. Li, D. Wang, H. He, L. Jiao, and Y. Xue, "Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems," *Neurocomputing*, vol. 249, pp. 48–63, Aug. 2017.
[31] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*. Berlin, Germany: Springer, 2004, pp. 492–508.
[32] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7386–7398, Nov. 2015.
[33] M. M. Azadjalal, P. Moradi, A. Abdollahpouri, and M. Jalili, "A trust-aware recommendation method based on Pareto dominance and confidence concepts," *Knowl.-Based Syst.*, vol. 116, no. 10, pp. 130–143, 2017.
[34] K. Ji and H. Shen, "Addressing cold-start: Scalable recommendation with tags and keywords," *Knowl.-Based Syst.*, vol. 83, pp. 42–50, Jul. 2015.
[35] C. Luo *et al.*, "Gaussian-Gamma collaborative filtering: A hierarchical Bayesian model for recommender systems," *J. Comput. Syst. Sci.*, 2017, doi: 10.1016/j.jcss.2017.03.007.
[36] X. Yang, Y. Guo, Y. Liu, and H. Steck, "A survey of collaborative filtering based social recommender systems," *Comput. Commun.*, vol. 41, pp. 1–10, Mar. 2014.
[37] J.-H. Liu, T. Zhou, Z.-K. Zhang, Z. Yang, C. Liu, and W.-M. Li, "Promoting cold-start items in recommender systems," *PLoS ONE*, vol. 9, no. 12, p. e113457, 2014.
[38] X. Zhao, "Research on top-N recommendation with collaborative filtering," Ph.D. dissertation, Dept. Comput. Sci. Technol., Beijing Inst. Technol., Beijing, China, Jun. 2014.
[39] C. Ma, "A guide to singular value decomposition for collaborative filtering," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, 2008, pp. 1–14. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.571.6274&rep=rep1&type=pdf
[40] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, 2007, pp. 5–8.
[41] L. Ren, "Research on some key issues of recommender systems," Ph.D. dissertation, Dept. Comput. Sci. Technol., East China Normal Univ., Shanghai, China, Mar. 2012.
[42] P. Parhi, A. Pal, and M. Aggarwal, "A survey of methods of collaborative filtering techniques," in *Proc. Int. Conf. Inventive Syst. Control*, Jan. 2017, pp. 1–7.
[43] X. Chang, Y.-L. Yu, and Y. Yang, "Robust top-k multiclass SVM for visual category recognition," in *Proc. ACM Knowl. Discovery Data Mining (SIGKDD)*, 2017, pp. 75–83.
[44] S. Huang, J. Ma, P. Cheng, and S. Wang, "A hybrid multigroup coclustering recommendation framework based on information fusion," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 2, p. 27, Mar. 2015.
[45] Q. Gao, L. Gao, J. Fan, and J. Ren, "A preference elicitation method based on bipartite graphical correlation and implicit trust," *Neurocomputing*, vol. 237, pp. 92–100, May 2017.
[46] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan, "User perception of differences in recommender algorithms," in *Proc. 8th ACM Conf. Recommender Syst. (RecSys)*, Silicon Valley, CA, USA, Oct. 2014, pp. 161–168.
[47] B. Huang, X. Yan, and J. Lin, "Collaborative filtering recommendation algorithm based on joint nonnegative matrix factorization," *Pattern Recognit. Artif. Intell.*, vol. 29, no. 6, pp. 725–734, Aug. 2016.
[48] D.-K. Chen and F.-S. Kong, "Hybrid Gaussian pLSA model and item based collaborative filtering recommendation," *Comput. Eng. Appl.*, vol. 46, no. 23, pp. 209–211, 2010.
[49] M. Fan, "Non-negative matrix factorization and clustering methods application research in personalized recommendation system," M.S. thesis, Dept. Comput. Technol., East China Jiaotong Univ., Nanchang, China, 2012.
[50] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. 1st ACM Conf. Recommender Syst.*, Minneapolis, MN, USA, 2007, pp. 17–24.
[51] H. Koohi and K. Kiani, "A new method to find neighbor users that improves the performance of collaborative filtering," *Expert Syst. Appl.*, vol. 83, pp. 30–39, Oct. 2017.
[52] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha, "Like like alike: Joint friendship and interest propagation in social networks," in *Proc. Int. Conf. World Wide Web*, 2011, pp. 537–546.
[53] F. Xia, N. Y. Asabere, A. M. Ahmed, J. Li, and X. Kong, "Mobile multimedia recommendation in smart communities: A survey," *IEEE Access*, vol. 1, pp. 606–624, Sep. 2013.
[54] Y.-T. Pan, F.-Z. He, and H.-P. Yu, "Social recommendation algorithm using implicit similarity in trust," *Chin. J. Comput.*, vol. 41, no. 1, pp. 65–81, 2018.
[55] A. Hernando, F. Ortega, and J. Tejedor, "Incorporating reliability measurements into the predictions of a recommender system," *Inf. Sci.*, vol. 218, pp. 1–16, Jan. 2013.
[56] J. Protasiewicz *et al.*, "A recommender system of reviewers and experts in reviewing problems," *Knowl.-Based Syst.*, vol. 106, pp. 164–178, Aug. 2016.
[57] X. Song, L. Nie, L. Zhang, M. Akbari, and T.-S. Chua, "Multiple social network learning and its application in volunteerism tendency prediction," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 213–222.
[58] M. K. Najafabadi, M. N. Mahrin, and S. Chuprat, H. M. Sarkan, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Comput. Hum. Behav.*, vol. 67, no. 2, pp. 113–128, 2017.
[59] L. Zhou, X. Ping, and S. Xu, "Clustering integration algorithm based on spectral clustering," *Acta Automat. Sinica*, vol. 38, no. 8, pp. 1335–1342, 2012.
[60] R. Chen *et al.*, "A hybrid recommendation method and development framework of user interface patterns based on hypergraph theory," *Int. J. Innov. Comput., Inf. Control*, vol. 13, no. 4, pp. 1169–1185, Aug. 2017.
[61] M. Jiang *et al.*, "Social contextual recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Oct. 2012, pp. 45–54.
[62] P. Moradi, S. Ahmadian, and F. Akhlaghian, "An effective trust-based recommendation method using a novel graph clustering algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 436, pp. 462–481, Oct. 2015.
[63] C. He, D. Parra, and K. Verbert, "Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities," *Expert Syst. Appl.*, vol. 56, no. 9, pp. 9–27, 2016.
[64] A. J. B. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *Proc. 9th ACM Conf. Recommender Syst. (RecCys)*, 2015, pp. 43–50.
[65] X. Zheng, Y. Luo, L. Sun, X. Ding, and J. Zhang, "A novel social network hybrid recommender system based on hypergraph topologic structure," *World Wide Web*, vol. 21, no. 4, pp. 985–1013, 2017.
[66] H. Feng and X. Qian, "Recommendation via user's personality and social contextual," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 1521–1524.
[67] M. Aharon, O. Anava, and N. Avigdor-Elgrabli, "ExcUseMe: Asking users to help in item cold-start recommendations," in *Proc. 9th ACM Conf. Recommender Syst. (RecCys)*, Vienna, Austria, Sep. 2015, pp. 83–90.
[68] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. 4th ACM Conf. Recommender Syst. (RecCys)*, 2010, pp. 39–46.
[69] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. World Wide Web*, vol. 4, 2001, pp. 285–295.
[70] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (RecCys)*, 2009, pp. 203–210.

[71] J. He and W. W. Chu, "A social network-based recommender system (SNRS)," in *Data Mining for Social Network Data*. Boston, MA, USA, Springer, 2010, pp. 47–74.

[72] M. Ramezani, P. Moradi, and F. Akhlaghian, "A pattern mining approach to enhance the accuracy of collaborative filtering in sparse data domains," *Phys. A, Stat. Mech. Appl.*, vol. 408, no. 32, pp. 72–84, 2014.

[73] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw. Anal. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[74] J. Li, C. Chen, H. Chen, and C. Tong, "Towards context-aware social recommendation via individual trust," *Knowl.-Based Syst.*, vol. 127, pp. 58–66, Jul. 2017.

[75] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.

[76] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.

[77] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.

[78] M. Nilashi, O. Ibrahi, and K. Bagherifard, "A recommender system based on collaborative filtering using Ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.

[79] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Mining Knowl. Discovery*, vol. 22, no. 3, pp. 493–521, 2011.

[80] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2007, pp. 1257–1264.

[81] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst. (RecCys)*, vol. 45, 2010, pp. 26–30.

[82] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.

[83] Z. Sun *et al.*, "Recommender systems based on social networks," *J. Syst. Softw.*, vol. 99, pp. 109–119, Jan. 2015.

[84] S. Chen, S. Owusu, and L. Zhou, "Social network based recommendation systems: A short survey," in *Proc. Int. Conf. Social Comput.*, Sep. 2013, pp. 882–885.

[85] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. DBLP*, 2011, pp. 287–296.

[86] Y. Wang, X. Wang, and Z. Wan-Li, "Trust prediction modeling based on social theories," *J. Softw.*, vol. 25, no. 12, pp. 2893–2904, 2014.

[87] Y. Cao, W. Li, and D. Zheng, "An improved neighborhood-aware unified probabilistic matrix factorization recommendation," *Wireless Pers. Commun.*, vol. 102, no. 4, pp. 3121–3140, 2018.

[88] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1267–1275.

[89] U. Liji, Y. Chai, and J. Chen, "Improved personalized recommendation based on user attributes clustering and score matrix filling," *Comput. Standards Interfaces*, vol. 57, no. 11, pp. 59–67, 2018.

[90] H. Li, D. Wu, and N. Mamoulis, "A revisit to social network-based recommender systems," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 1239–1242.

[91] L. Guo, J. Ma, Z. M. Chen, and H. R. Jiang, "Incorporating item relations for social recommendation," *Chin. J. Comput.*, vol. 37, no. 1, pp. 219–228, 2014.

[92] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. CIKM*, no. 10, Napa Valley, CA, USA, 2008, pp. 26–30.

[93] Y. Yu, Y. Gao, H. Wang, and S. Sun, "Integrating user social status and matrix factorization for item recommendation," *J. Comput. Res. Develop.*, vol. 55, no. 1, pp. 113–124, 2018.

[94] H. Li, X.-P. Ma, and J. Shi, "Incorporating trust relation with PMF to enhance social network recommendation performance," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 6, pp. 113–124, 2018.

[95] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM J. Comput. Cultural Heritage*, vol. 1, no. 1, pp. 1–35, Jul. 2017.

[96] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, Aug. 2009, Art. no. 421425, doi: 10.1155/2009/421425.

[97] I. Portugal, P. Alencar, and D. Cowan, "The use of machine learning algorithms in recommender systems: A systematic review," *Expert Syst. Appl.*, vol. 97, pp. 205–227, Dec. 2018.

[98] Y. Zuo, J. Zeng, M. Gong, and L. Jiao, "Tag-aware recommender systems based on deep neural networks," *Neurocomputing*, vol. 204, no. 1, pp. 51–60, 2016, doi: 10.1016/j.neucom.2015.10.134.

[99] W. Zhang, Y. Du, Y. Yang, and T. Yoshida, "DeRec: A data-driven approach to accurate recommendation with deep learning and weighted loss function," *Electron. Commerce Res. Appl.*, vol. 31, pp. 12–23, Oct. 2018.

[100] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, "Dual-regularized matrix factorization with deep neural networks for recommender systems," *Knowl.-Based Syst.*, vol. 145, pp. 46–58, Apr. 2016.

[101] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017.

[102] T. K. Paradarami, N. D. Bastian, and J. L. Wightman, "A hybrid recommender system using artificial neural networks," *Expert Syst. Appl.*, vol. 83, pp. 300–313, Oct. 2017.

[103] M. Luo, X. Chang, Z. Li, L. Nie, A. G. Hauptmann, and Q. Zheng, "Simple to complex cross-modal learning to rank," *Comput. Vis. Image Understand.*, vol. 163, pp. 67–77, Oct. 2017.

[104] Z. Li, F. Nie, X. Chang, and Y. Yang, "Beyond trace ratio: Weighted harmonic mean of trace ratios for multiclass discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2100–2110, Oct. 2017.

[105] X. Kong, M. Mao, and W. Wang, "VOPRec: Vector representation learning of papers with text information and structural identity for recommendation," *IEEE Trans. Emerg. Topics Comput.*, 2018, doi: 10.1109/TETC.2018.2830698.

[106] Y. Lu and J. Cao, "Research status and future trends of recommender systems for implicit feedback," *Comput. Sci.*, vol. 43, no. 4, pp. 7–15, 2016.

[107] A. M. Jorge *et al.*, "Scalable online top-N recommender systems," in *Proc. Int. Conf. Electron. Commerce Web Technol.*, vol. 278. Springer, 2017, pp. 3–20.

[108] F. Alqadah, C. K. Reddy, J. Hu, and H. F. Alqadah, "Biclustering neighborhood-based collaborative filtering method for top-n recommender systems," *Knowl. Inf. Syst.*, vol. 44, no. 2, pp. 475–491, 2015.

[109] T. Aytekin and M. Ö. Karakaya, "Clustering-based diversity improvement in top-N recommendation," *J. Intell. Inf. Syst.*, vol. 42, no. 1, pp. 1–18, 2014.

[110] H. Qiu, Y. Liu, G. Guo, Z. Sun, J. Zhang, and H. T. Nguyen, "BPRH: Bayesian personalized ranking for heterogeneous implicit feedback," *Inf. Sci.*, vol. 453, pp. 80–98, Jul. 2018.

[111] L. Liao, Y. Zhu, and F. Le, "Ranking recommendation combining trust and similarity for implicit feedback," *Appl. Res. Comput.*, vol. 35, no. 12, pp. 1–6, 2017.

[112] Y. Jian, Z.-S. Wang, L. Qi, and W.-J. Su, "Personalized recommendation based on large-scale implicit feedback," *J. Softw.*, vol. 25, no. 9, pp. 1953–1966, 2014.

[113] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[114] A. Karatzoglou, X. Amatriain, and L. Baltrunas, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. ACM Conf. Recommender Syst. (Recsys)*, Barcelona, Spain, Sep. 2010, pp. 79–86.

[115] L. Yao, Q. Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, and Q. He, "Context-aware point-of-interest recommendation using tensor factorization with social regularization," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2015, pp. 1007–1010.

[116] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, 2014.

[117] Y. Ying, L. Chen, and G. Chen, "A temporal-aware POI recommendation system using context-aware tensor decomposition and weighted HITS," *Neurocomputing*, vol. 242, pp. 195–205, Jun. 2017.

[118] Z. Zeng, Z. Li, D. Cheng, H. Zhang, K. Zhan, and Y. Yang, "Two-stream multi-rate recurrent neural network for video-based pedestrian re-identification," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3179–3186, Jul. 2018, doi: 10.1109/TII.2017.2767557.

[119] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1617–1632, Aug. 2017.

[120] Z. Zeng and Q. Lu, "Investigation of novel partitioned-primary hybrid-excited flux-switching linear machines," *IEEE Trans. Ind. Electron.*, vol. 65, no. 12, pp. 9804–9813, Dec. 2018.
[121] X. Chang, Z. Ma, Y. Yang, Z. Zeng, and A. G. Hauptmann, "Bi-level semantic representation analysis for multimedia event detection," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1180–1197, May 2017.
[122] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann, "Feature interaction augmented sparse learning for fast Kinect motion detection," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3911–3920, Aug. 2017.
[123] C. Gong, X. Chang, M. Fang, and J. Yang, "Teaching semi-supervised classifier via generalized distillation," in *Proc. IJCAI*, 2018, pp. 2156–2162.
[124] R. Pan *et al.*, "One-class collaborative filtering," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 502–511.
[125] L. Xiang, *Recommended System Practice*. Beijing, China: Posts & Telecom Press, 2012, pp. 51–77. [Online]. Available: http://vdisk.weibo.com/s/aSXlSkLAQjzMT
[126] B. Schölkopf, J. Platt, and T. Hofmann, "Learning to rank with nonsmooth cost functions," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 193–200.
[127] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, vol. 8, Dec. 2008, pp. 263–272.
[128] J. Delporte, A. Karatzoglou, T. Matuszczyk, and S. Canu, "Socially enabled preference learning from implicit feedback data," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 145–160.

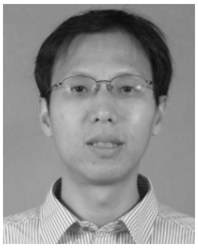
RUI CHEN received the B.S. degree in computer science and technology from the Henan Institute of Science and Technology, Xinxiang, China, in 2004, and the M.S. degree in computer application technology from Northwest University, Xi'an, China, in 2009, where he is currently pursuing the Ph.D. degree in computer application and technology. His research interests include human–computer interaction, recommender systems, social networks, and natural language processing.

QINGYI HUA received the B.S., M.S., and Ph.D. degrees in computer software and theory from Northwest University, Xi'an, China, in 1982, 1988, and 2006, respectively. Since 1998, he has been a Professor with the Human-Computer Interaction Laboratory, Northwest University. From 1992 to 1995 and from 2000 to 2004, he was involved in human–computer project research at the National Information Science Research Center, Germany. His research interests include human–computer interaction, recommender systems, and user interface engineering. He is a member of the Editorial Board of the *Chinese Journal of Computers* and Human-Computer Interaction Professional Committee.

YAN-SHUO CHANG received the Ph.D. degree from Northwest University in 2013. He is currently a Lecturer with the School of Information, Xi'an University of Finance and Economics. His current research interests include machine learning and its applications to multimedia analysis and quantitative trading.

BO WANG received the B.S. degree from the Xi'an University of Architecture and technology in 2003 and the M.S. degree from Xidian University in 2007. He is currently pursuing the Ph.D. degree with Northwest University. His main research interests include human–computer interaction and recommender systems.

LEI ZHANG received the B.S. degree in computer science and technology from Shanxi Normal University, Linfen, China, in 2003, the M.S. degree in computer application technology from Shanxi University, Taiyuan, China, in 2008, and the Ph.D. degree in computer application technology from Northwest University, Xi'an, China, in 2016. His research interests include image processing, pattern recognition, and machine learning.



XIANGJIE KONG (M'13–SM'17) received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China. He is currently an Associate Professor with the School of Software, Dalian University of Technology, China. He has published over 70 scientific papers in international journals and conferences (with over 50 indexed by ISI SCIE). His research interests include intelligent transportation systems, mobile computing, and cyber-physical systems. He is a Senior Member of CCF and a member of ACM. He has served as an (guest) editor for several international journals and the workshop chair or a PC member for a number of conferences.