Received September 12, 2017, accepted October 10, 2017, date of publication October 12, 2017,
date of current version November 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2762459

A Social Recommender Based on Factorization and Distance Metric Learning

JUNLIANG YU[1,2], MIN GAO[1,2], (Member, IEEE), WENGE RONG[3], YUQI SONG[1,2], AND QINGYU XIONG[1,2]

[1]Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing University, Ministry of Education, Chongqing 400044, China
[2]School of Software Engineering, Chongqing University, Chongqing 400044, China
[3]School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Corresponding author: Junliang Yu (yu.jl@cqu.edu.cn)

This work was supported in part by the Basic and Advanced Research Projects in Chongqing under Grant cstc2015jcyjA40049, in part by the National Natural Science Foundation of China under Grant 61472021, and in part by the Graduate Scientific Research and Innovation Foundation of Chongqing under Grant cys17035.

ABSTRACT Traditional recommender systems often suffer from the problem of data sparsity, because most users rate only a few of the millions of possible items. With the development of social platforms, incorporating abundant social relationships into recommenders can help to overcome this issue, because users' preferences can be inferred from those of their friends. Most existing social recommenders are based on matrix factorization, a collaborative filtering model that has been proven to be effective. In this paper, we introduce a novel social recommender based on the idea that distance reflects likability. Compared with matrix factorization, the proposed model enables us to obtain a spatial understanding of the latent factor space and how users and items are positioned inside the space by combining the factorization model and distance metric learning. In our method, users and items are initially mapped into a unified low-dimensional space. The positions of users and items are jointly determined by ratings and social relations, which can help to determine appropriate locations for users who have few ratings. Finally, the learned metrics and positions are used to generate understandable and reliable recommendations. Experiments conducted on real-world data sets have shown that compared with methods based on only matrix factorization, our method significantly improves the recommendation accuracy.

INDEX TERMS Social recommendations, distance learning, collaborative filtering, matrix factorization.

I. INTRODUCTION

The rapid development of the Internet has made information exchange convenient, but it has also led to a dilemma called information overload [1], [2]. For instance, facing an abundance of books in an online book store, how can a user select the book that best fits his taste in a limited time? To mitigate this problem, personalized customization services are urgently demanded. As an automatic system that can recommend an appropriate item, the recommender system plays an increasingly important role in facilitating information filtering. Typically, recommender systems are predicated on collaborative filtering (CF), a technique that depends on collective historical ratings to predict items that will be positively rated by the active user [3]. However, most users rate few of the millions of available items. The lack of common ratings results in a degradation of recommendation accuracy. Social recommender systems have emerged with the growing popularity of online social platforms [4]. These systems utilize both rating information and social relationships to generate recommendations, and the hypothesis behind them is that users are affected by their friends with respect to decision making [5]. Therefore, the preferences of users with few ratings can be inferred from the ratings of their friends. It has been proven that social recommender systems can generate more accurate recommendations, especially for so-called cold-start users [6].

The main approaches of CF can be categorized into two groups: memory-based methods and model-based methods [3]. In contrast to memory-based methods, model-based methods usually leverage previous ratings to train models in a more holistic manner; therefore, they are highly competitive, if not the best, and are widely adopted to build recommender systems [3]. Most existing recommender systems are based on matrix factorization [7], a basic model that has been extensively used owing to its scalability and efficiency. Generally, matrix factorization decomposes the user-item rating matrix into a user-preference matrix and an item-characteristics matrix based on the observed ratings.

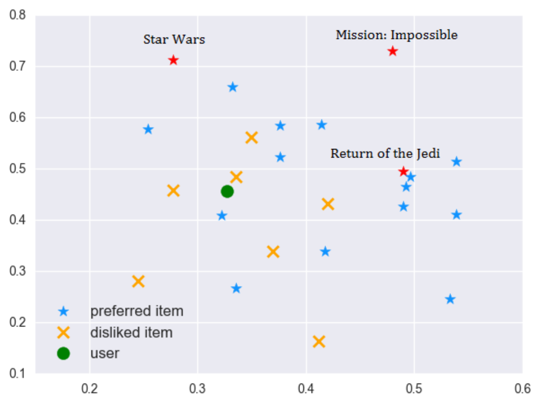


FIGURE 1. **An illustration of matrix factorization. The active user (green circle) searches a large area for desired items (stars) in the latent factor space.**

Each user and item is denoted with a low-dimensional vector called the latent factors. Then, the obtained latent factors are used to complement the rating matrix with their dot products, which can be considered as a type of interaction between users and items.

However, in most cases, matrix factorization fails to explain the interactions in a fine-grained view. For example, in Fig. 1, given latent factors derived from matrix factorization, a user and some items are embedded in a 2D latent factor space (data source: MovieLens). The stars represent the preferred items of the active user. Among them, the red stars are sorted out because they all belong to the same genre, "Action and Adventure", and all of them were positively rated by the active user and other users. However, despite the fact that they are similar, they are widely spread in the 2D latent space. According to the general explanation of matrix factorization, the latent factors of items represent the characteristics of items. But how can they be scattered with such large intervals? The explanation is therefore unconvincing.

In this paper, we propose a novel method, **SocialFD**, that is based on matrix factorization but has good interpretability. The method enables us to obtain a spatial understanding of the latent factor space and how users and items are positioned inside the space. The principle of the proposed method is that distance reflects likability; namely, in our model, the closer two points are (one is the user, the other is an item), the more likely is consumption or a click. Likewise, if two users are close, they are likely to be friends or potential friends. Previous research has introduced "distance" notion into CF [8]–[10]. But they usually adopted Euclidean distance to measure the gap among users and items. However, in our assumption, dimensionalities in the latent space are not orthogonal to each other, so we consider that Mahalanobis distance should be chosen instead. Existing research has proven that appropriately designed distance metrics can significantly benefit KNN classification accuracy compared to the standard Euclidean distance [11], [12]. It motivates us to integrate distance metric learning into CF by learning a symmetric positive semi-definite matrix to uncover the relation among different dimensionalities in latent space. As an important topic, distance metric learning [13] has been applied in many domains, including information retrieval, supervised classification, and clustering [14]–[16]. In contrast to general distance metric learning, our model needs to learn the samples (latent factors) and the distance metric simultaneously, which makes the model scalable. At the end of training, users are spatially close to their friends and preferred items and far away from their disliked items. The distance between users and items is jointly determined by ratings and social relations. That is to say, if a user has only a few ratings, their social connections can help to determine his location by pulling him near to his friends and recommending items around his friends to him. The experiments conducted on three real-world datasets show that our method manages to find proper locations for users and items, and significantly improves the quality of social recommendations.

The remainder of this paper is organised as follows. In section 2, the preliminaries are briefly introduced. Section 3 focuses on the proposed method - SocialFD. Section 4 reports the experimental results. In section 5, related works are introduced. Finally, in section 6 we conclude this paper and propose some potential future work.

II. PRELIMINARIES

A. TASKS OF SOCIAL RECOMMENDER SYSTEMS

In recommender systems, users can be defined as the set $\mathbf{U} = \{u_1, \ldots u_m\}$, and items can be defined as the set $\mathbf{I} = \{i_1, \ldots i_n\}$. Ratings given by users for items are marked with the matrix $\mathbf{R} = [r_{u,i}]_{m\times n}$, and $r_{u,i}$ denotes the rating from user u of item i. In reality, the density of the available ratings is often less than 1% [17]. To overcome the data sparsity problem, social information is incorporated. In a social network, each user has friends denoted by vector N_u. The user's missing ratings, to some degree, can be inferred from those of his friends. Regarding the edges in the social graph as trust statements, which are real numbers in [0, 1], we mark all trust statements from users with an adjacent matrix $\mathbf{T} = [t_{u,v}]_{m\times m}$. The task of a social recommender can be summarized as follows: given a user u and an item i, use the known information in $\mathbf{R}$ and $\mathbf{T}$ to predict $r_{u,i}$.

B. MATRIX FACTORIZATION

Generally, most model-based CF methods are based on matrix factorization [7], which maps both users and items into a low-dimensional latent-factor space such that user-item interactions are modeled as inner products in that space and user and item latent factors are denoted by $p_u \in \mathbb{R}^k$ and $q_i \in \mathbb{R}^k$, respectively. The resulting dot product, $p_u^T q_i$, can capture the interaction between user u and item i, which is used to predict the missing ratings.

Let N^R be an indicator of whether a rating has been observed in the user-item rating matrix R,

$$N^R_{u,i} = \begin{cases} 1 & \text{if } r_{u,i} \text{ is observed} \\ 0 & \text{if } r_{u,i} \text{ is missing} \end{cases} \tag{1}$$

Matrix factorization optimizes the following objective:

$$\mathcal{L} = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} N_{u,i}^{R}(r_{u,i} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2), \quad (2)$$

where the algorithmic parameter λ controls the magnitudes of the latent factors.

An extended model, called SVD [18], further considers biases as it assumes that much of the observed variation in ratings is due to effects associated with either users or items. For example, typical collaborative filtering data exhibits large systematic tendencies for some users to give higher ratings than others and for some items to receive higher ratings than others. Therefore, the optimization objective of SVD can be stated as:

$$\mathcal{L} = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} N_{u,i}^{R}(r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2 + \sum_{u=1}^{m} b_u + \sum_{i=1}^{n} b_i^2), \quad (3)$$

where μ represents the overall average rating, and b_u and b_i indicate the deviations of user u and item i.

C. DISTANCE METRIC LEARNING

Distance metric learning is crucial in real-world applications. Previous research [14]–[16] has shown that a good distance metric can remarkably improve the performance of learners that rely on spatial data. Generally, training samples for supervised distance metric learning are cast into pairwise constraints. Then, the supervised distance metric learning can be further divided into two subclasses: global distance metric learning and local distance metric learning [19]. In our work, we concentrate on applying the former to recommender systems. The target of global distance metric learning is to learn a distance metric subject to a given set of constraints in a global sense. Let the distance metric be denoted by matrix $A \in \mathbb{R}^{k\times k}$ and the distance between any two data points x and y be expressed by

$$d_A^2(x, y) = \|x - y\|_A^2 = (x - y)A(x - y)^T. \quad (4)$$

In Eq. 4, A has to be a positive semi-definite matrix to keep the distance non-negative and symmetric. The global optimization problem with constraints can be stated as

$$\min_{A\in\mathbb{R}^{k\times k}} \sum_{(x,y)\in S} \|x - y\|_A^2,$$
$$s.t.\ A \succeq 0, \quad \sum_{(x,y)\in D} \|x - y\|_A^2 \geq \beta, \quad (5)$$

where S denotes the set of equivalent constraints in which x and y belong to the same class, D denotes the set of inequivalent constraints in which x and y belong to different classes, and β is a constant that restricts the minimum distance between data points of different classes.

III. SOCIAL RECOMMENDER COMBINING FACTORIZATION AND DISTANCE METRIC LEARNING

Traditional recommender system techniques ignore the effect of social relations among users. With the rapid development of social networks, incorporating social information into recommender systems has become increasingly important. Most existing social recommender systems are based on matrix factorization. However, the meaning of the latent factors derived by matrix factorization are ambiguous. To make the recommendations reliable, we consider that latent factors should be more interpretable and intuitively understand. In this section, we propose a **S**ocial recommender that combines **F**actorization and **D**istance metric learning, called **SocialFD**.

A. MOTIVATION

The inspiration for SocialFD is that distance reflects likability. Enlightened by the success of distance metric learning on classification tasks [11], [12], we integrate distance metric learning with matrix factorization in our model. Recall that the main idea of distance metric learning is to learn a desired distance metric that can make data points with the same class label closer and discriminate data points in different sets with larger distance. However, SocialFD aims to minimize the distance between each user and his positively rated items and friends and to maximize the distance between each user and his negatively rated items. Consider the example in section 1; for the matrix factorization model, the desired items for the active user may be spread over a large area, which results in a failure to demonstrate the meaning of the latent factors. By contrast, for our model, the desired items are positioned in a smaller user-centered area, which is not only apt to capture finer-gained preferences but is also more intuitively understandable.

B. TRAINING PROCEDURE

In contrast to general distance metric learning, the samples in our models are the latent factors, which are not initially prepared, and the labels are classified into two types (like and dislike) according to the ratings expressed by users. Note that we consider higher scores to indicate positive ratings and lower scores to represent negative ratings. The sets of pairwise constraints are constructed as follows: given a user and an item, if the user positively rates the item, the pair is added to the set of equivalent constraints; otherwise, they are added to the set of inequivalent constraints.

To achieve the goal of our model, we need to simultaneously train the latent factors and the distance metric. The overall process is depicted in Fig. 2. First, we initialize two k-rank matrices filled with random values denoting the user latent matrix and the item latent matrix and define a matrix $\in \mathbb{R}^{k\times k}$ as the distance metric. The Mahalanobis distance between the users and items can be calculated as the products of the differences in latent factors and the distance metric. During the training stage, constraints are imposed to guarantee that users are spatially close to their friends and their preferred

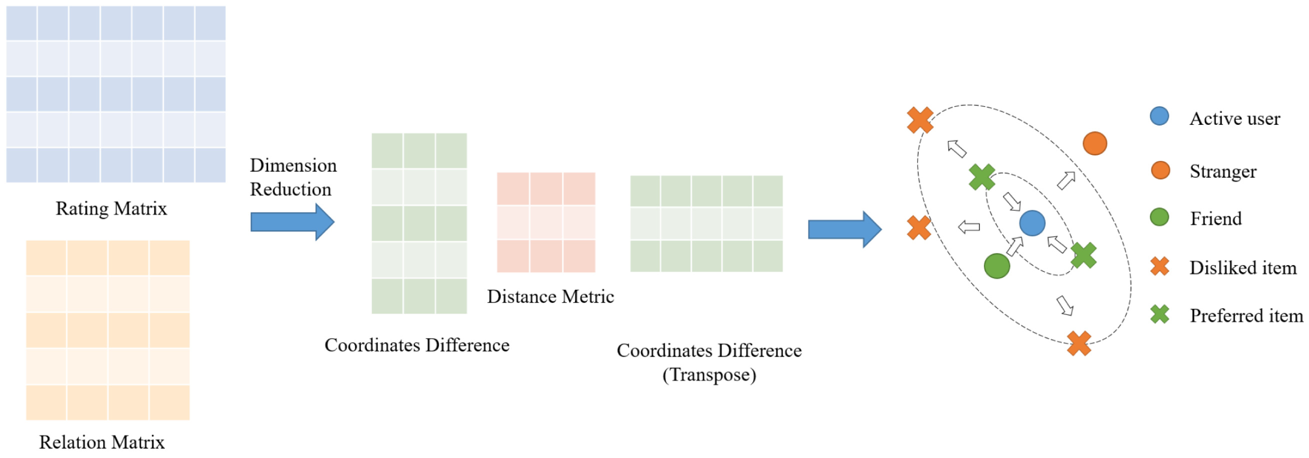


FIGURE 2. An illustration of the SocialFD model. Users (circles) and items (crosses) are embedded in low-dimensional space. The closer the two symbols are, the more likely that the user prefers the item/user.

items and far from their disliked items. The positions of users and items are jointly determined by the ratings and social relations. It means, if a user has only a few ratings, their social connections can help to determine his location by pulling him near to his friends and recommending items around his friends to him. Finally, the obtained latent factors can be interpreted as coordinates in the low-dimensional space, and the distance can be used to generate understandable recommendations.

C. MODEL FORMULATION

According to the description above, in SocialFD, the predicted rating is defined as

$$\hat{r}_{ui} = \mu + b_u + b_i - \|x_u - y_i\|_A^2, \tag{6}$$

where x_u and y_i denote the latent factors of user u and item i, respectively, $A \in \mathbb{R}^{k\times k}$, and k equals the dimensionality of x and y, which is much smaller than the dimensionality of the original rating matrix. Instead of learning a positive semi-definite matrix A, $H \in \mathbb{R}^{k\times k}$ can be learned with $A = HH^T$. Furthermore, H need not be positive semi-definite, which allows the problem to be solved with generic approaches. Hence, we can learn latent factors and the distance metric by solving the following optimization problem:

$$\begin{aligned}\mathcal{L} = &\frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n} N^R_{u,i}(r_{ui} - \mu - b_u - b_i + \|x_u - y_i\|_A^2)^2 \\ &+ \frac{\lambda}{2}(\sum_{u=1}^{m} b_u^2 + \sum_{u=1}^{m} b_i^2) + \frac{\eta}{2}\{\alpha \sum_{(u,i)\in P} \|x_u - y_i\|_A^2 \\ &+ (1-\alpha)\sum_{u\in\mathbf{U}}\sum_{v\in N_u} \|x_u - x_v\|_A^2 \\ &+ \alpha \sum_{(u,i)\in N} [1 - \|x_u - y_i\|_A^2]_+\},\end{aligned} \tag{7}$$

where P is the set of pairs containing user u and his positively rated items, N is the set of pairs containing user u and his negatively rated items, the last three terms are constraints used to adjust the user-user and user-item distance to an appropriate range, $[z]_+ = max(z; 0)$ is the standard hinge loss, λ controls the magnitudes of the biases, η is a parameter used to control the influence of the constraints, and α is a trade-off between the effects of the user-user distance and the user-item distance.

Many optimizers are available to identify good solutions for this problem [13], [20], [21]. In our work, we use stochastic gradient descent because it works very efficiently in the case of redundant data. A local minimum of the objective function given by Eq. 7 can be found by performing gradient descent in $b_u, b_i, x_u, y_i,$ *and* H,

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial b_u} &= \lambda b_u - e_{ui}\frac{\partial\mathcal{L}}{\partial b_i} = \lambda b_i - e_{ui} \\ \frac{\partial\mathcal{L}}{\partial y_i} &= -(e_{ui} \pm \alpha\eta)(x_u - y_i)W \\ \frac{\partial\mathcal{L}}{\partial x_u} &= -\eta(1-\alpha)\sum_{v\in N_u}(x_u - x_v)W \\ &\quad + (e_{ui} \pm \alpha\eta)(x_u - y_i)W \\ \frac{\partial\mathcal{L}}{\partial H} &= \eta(1-\alpha)\sum_{v\in N_u} H(x_u - x_v)^T(x_u - x_v) \\ &\quad + (e_{ui} \pm \alpha\eta)H(x_u - y_i)^T(x_u - y_i),\end{aligned} \tag{8}$$

where e_{ui} is the error between the observed rating and the predicted rating, γ is the learning rate, $W = (HH^T + H^T H)$, and the exact operator of the plus-minus sign is determined by the sign of the term related to distance in Eq. 7.

D. COMPLEXITY

With regard to complexity,compared with matrix factorization, SocialFD needs to learn only an additional distance metric matrix with $k \times k$ elements. The model can be trained off-line. For prediction, if we consider the binary operation as the basic operation, and the term $x_u - y_i$ is pre-computed, for each tuple, our model requires $(2k-1)k$ more operations than the matrix factorization model, which can be marked as $O(k^2)$. However, because k is a small number, which is consistent with the dimensionality of the latent factors of users and items, the cost is not be computationally expensive.

E. PROPAGATION OF DISTANCE

The notion of similarity propagation or trust propagation is closely related to CF [22]. Likewise, the propagation of distance in our model plays an important role. Given the information "user u likes item i and has a friend user k," SocialFD not only pulls user u and item i closer but also pulls user k and item i relatively close to one another. If user k has few rated items, the propagation of distance is very helpful to overcome the data sparsity problem by recommending user u' preferred items to user k. Similarly, SocialFD keeps user u far from the disliked item j, which indirectly pushes item j away from user k. Moreover, SocialFD also pulls users with similar preferences or indirect connections together. Therefore, the propagation of distance can help to discover potential desired items and friends.

F. INTEGRATING USER/ITEM FEATURES

One drawback of the matrix factorization model is that it is difficult to directly integrate the well-trained vector. By contrast, for SocialFD, an outstanding advantage is the flexible incorporation of the ready-made representation of the additional knowledge. In previous sections, we assumed that all collected data are ratings and social connections; however, in real scenarios, the profiles of users and items consist of abundant texts that can also be collected to help to improve the quality of the recommendations [23], [24]. For example, in some content-based systems, users can fuse their preferences into their reviews and user profiles, and items can show characteristics in their descriptions. Therefore, NLP models, such as word2vec [25], FastText [26], and WordRank [27], can be used to generate distributed representations for the opinions and descriptions. Then, the semantic-included representations can replace the initial random vectors x and y in SocialFD or be concatenated with random vectors. The new representations may help to learn a better distance metric for SocialFD as they contain more characteristics of users and items; furthermore, they are pre-trained, which can reduce the training time of SocialFD.

IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results. Three types of experiments are conducted: (1) the recommendation quality of the baselines and SocialFD is compared; (2) whether the change in distance in SocialFD is desired is confirmed; and (3) the sensitivity of the parameters is investigated.

A. EXPERIMENTAL SETUP

In this subsection, we introduce the datasets, the evalution metrics, and the baselines used to compare with SocialFD.

TABLE 1. Dataset statistics.

Dataset	#Users	#Items	#Ratings	#Relations	Rating Scale
Douban	2,848	39,586	894,887	35,770	1.0 - 5.0
Epinions	40,163	139,738	664,823	487,182	1.0 - 5.0
Ciao	7,375	105,114	284,086	111,781	1.0 - 5.0

Three common real-world datasets, Douban, Epinions and Ciao are used in our experiments. The statistics of the datasets released by [4], [28], and [29] are shown in Table 1.

In our experiments, root mean square error (RMSE) is chosen to measure the prediction error for all methods. Additionally, we use the ranking-based metrics Recall@k and MAP@k to measure the quality of the recommendation list. Recall@k assumes that all items ranked within the top-k are equivalent, and MAP (mean average precision) uses a monotonically increasing discount to emphasize the importance of higher ranks versus lower ranks.

RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{u,i}(r_{ui} - \hat{r}_{ui})^2}{N}}, \tag{9}$$

where N denotes the number of ratings in the test set. A lower RMSE indicates that missing ratings are predicted more precisely.

Recall@k is defined as

$$Recall@k = \frac{\sum_{u \in U} Hits_u / Rated_u}{|U|}, \tag{10}$$

where $Hits_u$ denotes the number of recommended items that appear in both the recommendation list and the rated-items list.

MAP@k is defined as

$$MAP@k = \frac{\sum_{u \in U} \sum_{n=1}^{k}(P(n)/min(m,k))}{|U|}, \tag{11}$$

where $P(n)$ represent the precision at cut-off n in the item list, and m is the length of the rated-items list of user u.

To demonstrate the performance improvement of SocialFD, we compare our method with the following methods.

- **PMF**: this method was proposed by Salakhutdinov and Minh [30] and is based on probabilistic matrix factorization.
- **SoRec**: this method was proposed in [31]. It is a social trust-aware recommendation method that factorizes the user-item rating matrix and users social trust network by sharing the same user latent space.
- **SocialMF**: this method was proposed in [22]. It forces the user's preferences to be similar to those of their friends, which is similar to SocialFD.
- **RSTE**: this method was proposed by Ma *et al.* [32]. It is a linear combination of the basic matrix factorization approach and a social-network-based approach.
- **SREE**: this method was proposed by Li *et al.* [10]. It is the most relevant work to ours, which integrates distance

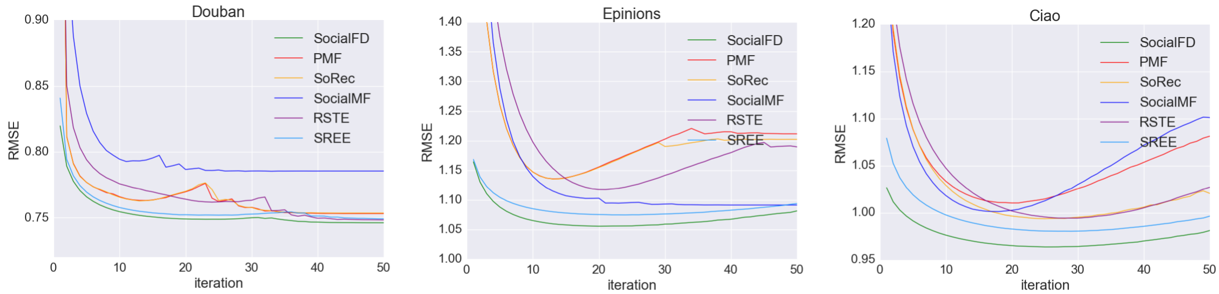


FIGURE 3. The RMSEs of all methods as the stochastic gradient descent algorithm proceeds.

TABLE 2. Best RMSEs of all methods.

Dataset	PMF	SoRec	SocialMF	RSTE	SREE	**SocialFD**
Douban	0.7541	0.7534	0.7852	0.7483	0.7491	**0.7461**
Epinions	1.1356	1.1352	1.0912	1.1174	1.0747	**1.0555**
Ciao	1.0104	0.9937	1.0012	0.9943	0.9804	**0.9638**

into CF and utilizes social connections as well. However, this method uses Euclidean distance to measure the gap.

To tune the methods, we use 80% of the data as the training set, from which we randomly select 10% as the validation set. For the remaining 20% of the data, we consider the users and items that appear in the training and validation sets to obtain the test set. We record the best parameters of these methods according to their performance on the validation set. Afterwards, all the experiments are performed with 5-fold cross validation.

B. PERFORMANCE FOR PREDICTING MISSING RATINGS

The prediction of missing ratings in the rating matrix is the primary goal of most recommendation methods. In this section, we compare the performance of SocialFD with that of the baselines. We set the step size $\gamma = 0.005$, $\alpha = 0.6$, and $\eta = 0.3$ for SocialFD. The dimensionality d of the latent factors and the distance metric is set to 20, and the regularization parameter λ is set to 0.01 for all the methods.

Fig. 3 shows the change in RMSE as the stochastic gradient descent algorithm proceeds. We can clearly observe that SocialFD outperforms the baselines. It should be noted that the predicted rating equation of SocialFD includes the overall average rating; it can obtain better initial predictions and therefore converges rapidly. In Table 2, we list the best RMSE for all the methods. Despite the fact that SocialFD is not fine tuned, it beats other methods by fairly large margins. Compared with PMF, SoRec, SocialMF, RSTE and SREE, SocialFD lowers the RMSE by 1.06%, 0.96%, 4.97%, 0.29% 0.40% on Douban, by 7.05%, 7.02%, 3.27%, 5.53%, and 1.76% on Epinions, and by 4.65%, 3.00%, 3.79%, 3.06%, and 1.69% on Ciao, respectively.

In reality, recommendations for users are usually presented as a recommendation list. Thus, we prefer the ranking metrics to RMSE. In the preprocessing stage, we binarize the explicit rating data by assigning 1 to the ratings not less than 4 and assigning 0 to ratings less than 4. In SocialFD, we use the distance $\|x_u - y_i\|_A^2$ to generate ranking scores for users on all items. By sorting the ranking scores in ascending order, we can obtain the recommendation lists. Table 3 shows the lists measured by the ranking metrics. As the users rate few of the items and the candidate items are numerous, the figures may seem small; however, this problem is inevitable and can reflect the capability of different methods. In Table 3, we can see that SocialFD is superior to the other methods on almost all the metrics. Compared with PMF, SoRec, SocialMF, RSTE and SREE, SocialFD improves Recall@50 by 66.43%, 63.88%, 31.78%, 24.28%, and −4.18% on Douban, by 28.70%, 43.08%, 16.95%, 15.94%, and 17.98% on Epinions, and by 57.43%, 43.52%, 55.52%, 31.56%, and 13.01% on Ciao, respectively. In particular, the MAP of SocialFD is significantly better than that of the other methods on all the datasets, which means that the preferred items will be ranked higher.

It should be noted that, PMF, SoRec, SocialMF, RSTE are matrix factorization based only methods, and SREE integrates Euclidean embedding into matrix factorization. The results show that SocialFD is superior to all of them, which validates our hypothesis that incorporating distance metric learning into CF can further improve the recommendation quality and Mahalanobis distance performs better than Euclidean distance in recommendation tasks.

C. CHANGE OF THE DISTANCE

SocialFD aims to place users spatially close to their friends and preferred items and far from their disliked items. In SocialFD, constraints for the distance are binding to help reach the objective. In this section, we confirm whether the obtained distance is desired. The settings for SocialFD are the same as those in section 4.2, and the experiments are conducted on the same two datasets.

Fig. 4 shows the variation in the distance curves. We call the distance between a user and his positively rated items *positive distance*. Similarly, *negative distance* is defined as the distance between a user and his negatively rated items. We use the distance ratio to denote the ratio of the negative

TABLE 3. The ranking quality of all methods.

	Douban				Epinions				Ciao			
Metric	Rec@20	Rec@50	MAP@50	MAP@100	Rec@20	Rec@50	MAP@50	MAP@100	Rec@20	Rec@50	MAP@50	MAP@100
PMF	2.18%	2.89%	0.0036	0.0030	1.48%	2.09%	0.0022	0.0024	2.64%	3.31%	0.0056	0.0064
SoRec	2.21%	2.94%	0.0038	0.0030	1.65%	1.88%	0.0030	0.0021	2.91%	3.63%	0.0059	0.0053
SocialMF	2.81%	3.65%	0.0045	0.0044	1.55%	2.30%	0.0027	0.0036	2.71%	3.35%	0.0049	0.0064
RSTE	3.26%	3.87%	0.0073	0.0058	1.78%	2.28%	0.0020	0.0033	3.13%	3.96%	0.0063	0.0061
SREE	**3.75%**	**5.02%**	0.0089	0.0113	1.67%	2.32%	0.0035	0.0038	3.56%	4.61%	0.0078	0.0081
SocialFD	3.63%	4.81%	**0.0091**	**0.0121**	**2.01%**	**2.69%**	**0.0042**	**0.0043**	**3.93%**	**5.21%**	**0.0088**	**0.0092**

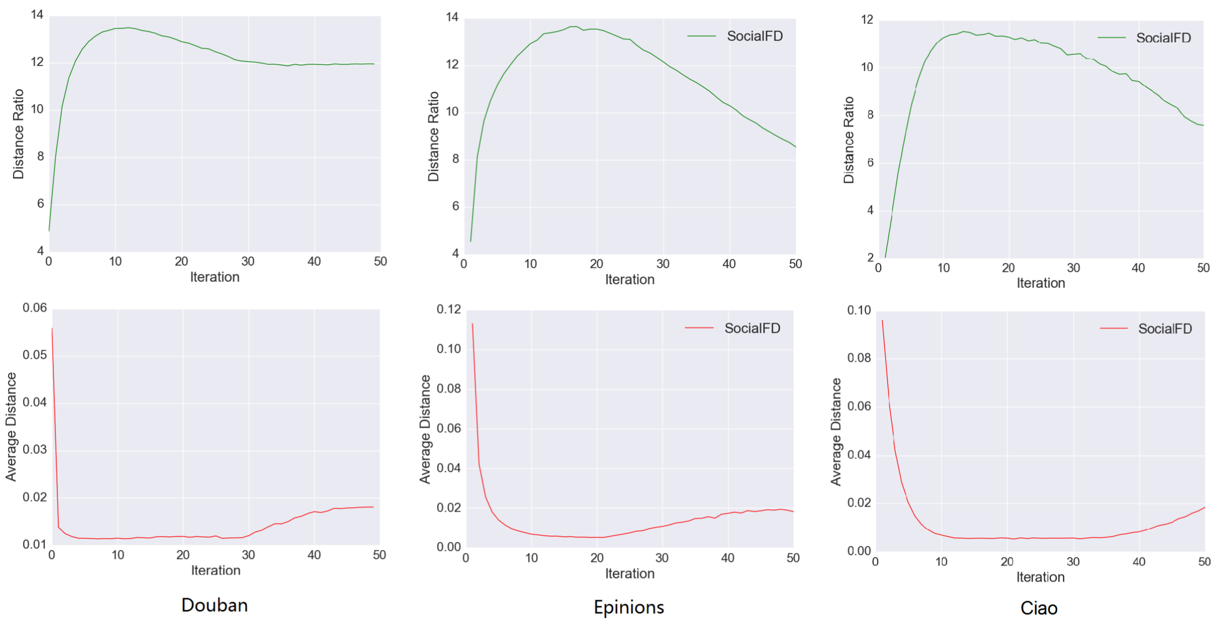


FIGURE 4. The variation of the distance as the stochastic gradient descent algorithm proceeds.

distance to the positive distance, which is defined as follows:

$$Distance\ Ratio = \frac{\sum_{(u,i)\in N} \|x_u - y_i\|_A^2}{\sum_{(u,i)\in P} \|x_u - y_i\|_A^2}. \tag{12}$$

The curves (green) in this section initially increase exponentially. After 30 iterations, the curve for Douban gradually reaches a stable state. Finally, the distance ratio converges to approximately 12. However, the curves for Epinions and Ciao start to decrease after 20 iterations and continues decreasing until the end. Moreover, the change in the distance curves is opposite to the change in the RMSE curves in Fig. 3. It appears that a higher distance ratio corresponds to a lower RMSE. In Fig. 3, with the overfitting of SocialFD, the distance ratio in Fig. 4 decreases obviously and persistently. We believe the connection beween the RMSE and the distance further proves that the motivation of our model is promising. The right part of Fig. 4 shows the change in the average distance among users with social connections. The average distance is defined as:

$$Average\ Distance = \frac{\sum_{u\in \mathbf{U}} \sum_{v\in N_u} \|x_u - x_v\|_A^2}{\sum_{u\in \mathbf{U}} |N_u|}. \tag{13}$$

As the iterations proceed, the average distance for both datasets decreases; however, when overfitting occurs in Fig. 3, the average distance starts to increase. This change indicates that a smaller distance between friends will lead to a more accurate result. Table 4 shows the connection between the RMSE and the change in distance more intuitively; it almost presents a negative correlation. So far, we can draw the conclusion that SocialFD achieves the goal and finds appropriate locations for users and items.

D. INVESTIGATION OF THE PARAMETER SENSITIVITY

In SocialFD, two hyper-parameters, η and α, are introduced. These two parameters are used to control the influence of the constraints and to balance the impact of user-item distance and user-user distance. In this subsection, we investigate the sensitivity of η and α. First, we fix $\alpha = 0.6$ and increase the value of η from 0 to 1 to observe the variation in the performance. Then, we fix $\eta = 0.3$ to observe the influence of changing α. The experiments are conducted on both datasets, and the other settings are the same as those in section 4.2.

The top half of Fig. 5 shows the variation of the RMSE for different values of η. SocialFD achieves the best performance

TABLE 4. The connection between the RMSE and the distance ratio.

		Iteration										
		1	5	10	15	20	25	30	35	40	45	50
Douban	RMSE	1.164	1.087	1.065	1.057	1.055	1.056	1.059	1.062	1.067	1.074	1.081
	Distance Ratio	4.870	12.071	12.918	13.368	13.364	13.001	12.592	12.072	11.939	11.936	11.952
	Average Distance	0.055	0.011	0.011	0.012	0.012	0.012	0.012	0.014	0.017	0.018	0.018
Epinions	RMSE	1.164	1.087	1.065	1.057	1.055	1.056	1.059	1.062	1.067	1.074	1.081
	Distance Ratio	4.542	11.151	12.918	13.504	13.527	13.093	12.146	11.268	10.293	9.340	8.539
	Average Distance	0.113	0.014	0.007	0.005	0.005	0.007	0.010	0.014	0.017	0.018	0.018
Ciao	RMSE	1.026	0.991	0.976	0.968	0.965	0.963	0.964	0.966	0.970	0.975	0.981
	Distance Ratio	2.045	8.381	11.270	11.375	11.293	11.039	10.584	10.056	9.421	8.431	7.576
	Average Distance	0.095	0.0151	0.006	0.005	0.005	0.005	0.005	0.006	0.0009	0.013	0.018

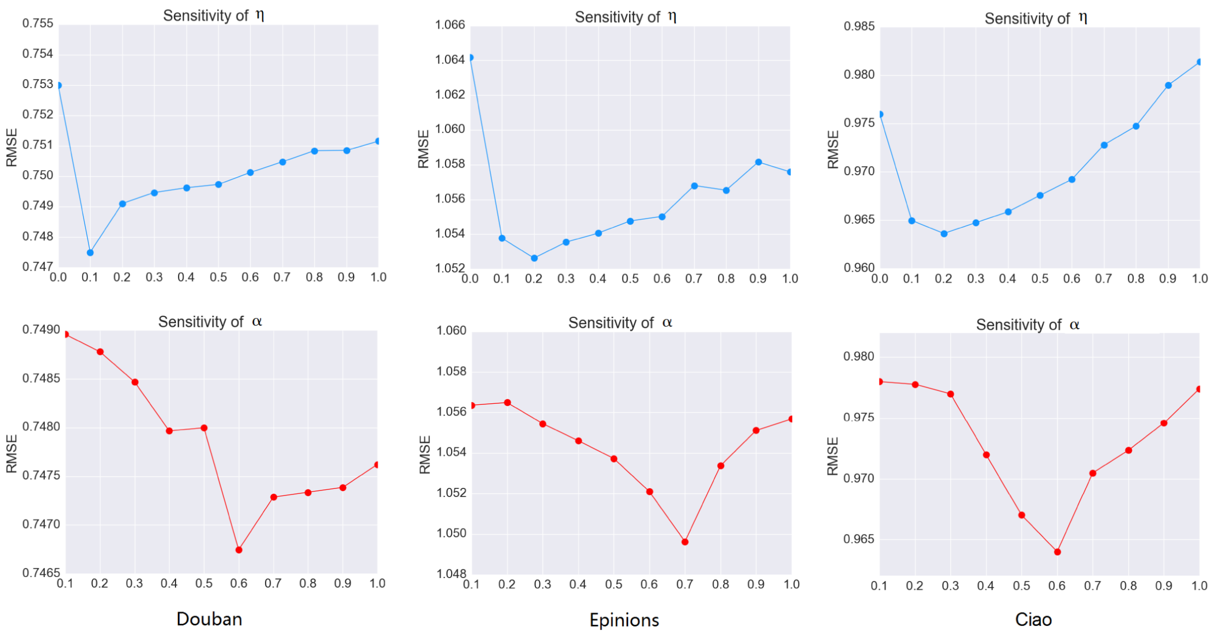


FIGURE 5. The sensitivity of η and α on Douban, Epinions and Ciao.

when $\eta = 0.1$ on Douban, and $\eta = 0.2$ on Epinions and Ciao. When $\eta = 0$, no constraints are imposed, but the worst performance is achieved, demonstrating that fusing the distance and using constraints to maintain the desired distance can improve the recommendation accuracy. The bottom part of Fig. 3 shows that SocialFD achieves the best performance when $\alpha = 0.6$ on Douban and Ciao, and $\alpha = 0.7$ on Epinions, while neither smaller values nor larger values lead to obvious performance degradation. This demonstrates that combining social relaitons and ratings can result in the best performance and that an appropriate user-item distance is more important than the user-user distance.

V. RELATED WORK

Complementing matrix factorization with further data, e.g., implicit feedback, temporal information or predefined metadata, has widely been accepted to increase algorithmic accuracy. Because of the convenience of matrix factorization for incorporating social information, several social recommenders based on matrix factorization have been proposed recently [10], [22], [31]–[38]. In contrast to rating-based recommenders, social recommenders capture social information as complementary input. Thus, in social recommender systems, the interactions among users have to be explored.

In [32], the authors proposed an ensemble method that considers the preference of users to be determined by their own tastes and their friends' tastes. Therefore, the missing ratings of a given user are predicted as a linear combination of the ratings of the user and his friends. Reference [31] proposed a model to co-factorize the rating matrix and the relation matrix into three low-rank matrices. In this model, social information and rating information are connected through the shared user latent feature space. In [36], two trust models were proposed. In contrast to other social recommenders, this work considers that both the trustors and trustees can affect users' preferences. Reference [37] introduced a model based on Bayesian inference for online social networks. In this method, the similarity value between each pair of the users is measured by using a set of conditional probabilities derived from their mutual ratings.

Among the existing research, the principle of [22], [33], and [38] is similar to that of our method. The authors of [22] proposed a method named SocialMF. The idea behind SocialMF is that a user's preferences should be similar to those of his friends. Thus, for a given user, the method forces the latent factors to be close to that of his friends. Similarly, in [38], the authors introduced a method that incorporates the overlapping community regularization into the matrix factorization framework. It considers that user's latent factors should be close to the average latent user factors of the communities to which he belongs. In [33], a method was proposed that takes both trust and distrust into consideration. The authors deemed that each user's latent factors should be similar to those of his friends and differ widely from those of distrusted users. However, these methods still regard latent factors as a representation of characteristics that lack of explicit meaning, which makes these methods less reliable.

Some existing research integrates distance into CF [8], [10]. The idea of [8] is that all items and users can be embedded in a unified Euclidean space, and the learned latent factors of users and items should reflect the negative correlation between the distance and the given rating.Therefore, the higher a given rating is, the closer the user and the item are spatially. In [10], the authors extended the model in [8], integrating social connections with ratings. The positions of users and items are jointly determined by the ratings and social relations, which makes this work highly relevant to ours. However, whether the Euclidean distance can precisely measure the gap between users and items is questionable. In addition, no constraints are imposed to guarantee that the obtained distance is desired, and it all depends on the weak correlation with the ratings, which may include some noise.

In [9], an item-ranking-based recommender was proposed. This model learns a joint metric space to encode not only users' preferences but also the user-user and item-item similarity. The authors focus on the collaborative filtering problem for implicit feedback and use collective metric learning to capture the interactions among all points in the metric space. They also demonstrate their model's ability to integrate various types of item features and prove that their model can uncover the fine-grained relationships among users' preferences. However, this methods do not consider social connections, resulting in failure when providing recommendations to users with few ratings. And also this model does not learn a metric matrix to reveal the relations among different dimensionalities.

VI. CONCLUSION

This paper is motivated by the assumption that distance can reflect likability. Based on this intuition, we proposed a novel recommendation method called SocialFD that combines distance metric learning and matrix factorization. The method aims to embed users and items into a unified space and to place users spatially close to their friends and their liked items and far from their disliked items. The positions of users and items are jointly determined by ratings and social relations, which can help to determine appropriate locations for users who have few ratings. The obtained distances are used to generate understandable and reliable recommendations.

In this paper, although we utilize social connections to provide user recommendations, we do not consider distrust. In general, negative opinions contain more information than positive opinions. Adding distrust to the distance-based social recommender can help to locate users more precisely. We will explore this potential direction in the future. In addition, we note that few works combine distance metric learning with link prediction. We consider that the idea of this paper can be extended to link prediction.

REFERENCES

[1] A. R. Lee, S.-M. Son, and K. K. Kim, "Information and communication technology overload and social networking service fatigue: A stress perspective," *Comput. Human Behavior*, vol. 55, pp. 51–61, Feb. 2016.
[2] D. J. Levitin, *The Organized Mind: Thinking Straight in the Age of Information Overload*. New York, NY, USA: Dutton, 2014.
[3] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.
[4] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Recommender Syst.*, 2007, pp. 17–24.
[5] R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *Proc. DELOS-NSF Workshop Personalization Recommender Syst. Digit. Libraries*, 2001, pp. 1–6.
[6] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw. Anal. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.
[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[8] M. Khoshneshin and W. N. Street, "Collaborative filtering via Euclidean embedding," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 87–94.
[9] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 193–201.
[10] W. Li *et al.*, "Social recommendation using Euclidean embedding," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 589–595.
[11] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Neural Inf. Process. Syst.*, 2004, pp. 513–520.
[12] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.
[13] B. Kulis, "Metric learning: A survey," *Found. Trends Mach. Learn.*, vol. 5, no. 4, pp. 287–364, 2013.
[14] P. Wu, S. C. H. Hoi, P. Zhao, C. Miao, and Z.-Y. Liu, "Online multi-modal distance metric learning with application to image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 454–467, Feb. 2016.
[15] B. Nguyen, C. Morell, and B. D. Baets, "Supervised distance metric learning through maximization of the Jeffrey divergence," *Pattern Recognit.*, vol. 64, pp. 215–225, Apr. 2017.
[16] J. Wang *et al.*, "Distance metric learning for soft subspace clustering in composite kernel space," *Pattern Recognit.*, vol. 52, pp. 113–134, Apr. 2016.
[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. World Wide Web*, 2001, pp. 285–295.
[18] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, 2010.
[19] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, Tech. Rep. 2, 2006, vol. 2.

[20] Q. Qian, R. Jin, J. Yi, L. Zhang, and S. Zhu, "Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD)," *Mach. Learn.*, vol. 99, no. 3, pp. 353–372, 2015.

[21] J. Hocke and T. Martinetz, "Global metric learning by gradient descent," in *Proc. Int. Conf. Artif. Neural Netw.*, 2014, pp. 129–135.

[22] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[23] C. Musto, "Enhanced vector space models for content-based recommender systems," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, Barcelona, Spain, Sep. 2010, pp. 361–364.

[24] G. Semeraro, P. Lops, P. Basile, and M. de Gemmis, "Knowledge infusion into content-based recommender systems," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA, Oct. 2009, pp. 301–304.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[26] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 427–431.

[27] S. Ji, H. Yun, P. Yanardag, S. Matsushima, and S. V. N. Vishwanathan, "Wordrank: Learning word embeddings via robust ranking," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 658–668.

[28] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.

[29] J. Tang, H. Gao, and H. Liu, "mTrust: Discerning multi-faceted trust in a connected world," in *Proc. Int. Conf. Web Search Web Data Mining (WSDM)*, Seattle, WA, USA, Feb. 2012, pp. 93–102.

[30] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[31] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.

[32] 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.*, 2009, pp. 203–210.

[33] H. Ma, M. R. Lyu, and I. King, "Learning to recommend with trust and distrust relationships," in *Proc. ACM Conf. Recommender Syst.*, 2009, pp. 189–196.

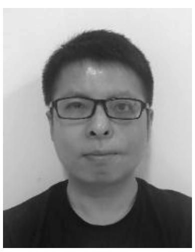
[34] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[35] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 123–129.

[36] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2016.

[37] J. Liu, C. Wu, and W. Liu, "Bayesian probabilistic matrix factorization with social relations and item contents for recommendation," *Decision Support Syst.*, vol. 55, no. 3, pp. 838–850, 2013.

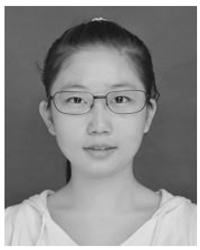
[38] H. Li, D. Wu, W. Tang, and N. Mamoulis, "Overlapping community regularization for rating prediction in social recommender systems," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 27–34.



JUNLIANG YU received the B.S. degree from Chongqing University, Chongqing, China, in 2013, where he is currently pursuing the master's degree with the School of Software Engineering. His research interests include recommender systems, anomaly detection, and cloud services.

MIN GAO (M'16) received the M.S. and Ph.D. degree in computer science from Chongqing University in 2005 and 2010, respectively. She was a Visiting Researcher with the School of Business, University of Reading. She is currently an Associate Professor with the School of Software Engineering, Chongqing University. Her research interests include recommendation systems, service computing, and data mining. She has authored over 30 refereed journal and conference papers in these areas. She has grants from the National Natural Science Foundation of China, the China Postdoctoral Science Foundation, and the China Fundamental Research Funds for the Central Universities. She is member of the CCF.

WENGE RONG received the B.Sc. degree from the Nanjing University of Science and Technology, China, in 1996, the M.Sc. degree from the Queen Mary College, University of London, U.K., in 2003, and the Ph.D. degree from the University of Reading, U.K., in 2010. He is currently an Associate Professor with Beihang University, China. He has many years of experience as a Senior Software Engineer in numerous research projects and commercial software products. His area of research includes machine learning, data mining, service computing, and information management.



YUQI SONG received the B.S. degree from Chongqing University, Chongqing, China, in 2016, where she is currently pursuing the master's degree with the School of Software Engineering, Chongqing University. Her research interests include recommender systems and anomaly detection.

QINGYU XIONG received the B.S. and M.S. degree from the School of Automation, Chongqing University, in 1986 and 1991, respectively, and the Ph.D. degree from the Kyushu University of Japan in 2002. He is currently the Dean of the School of Software Engineering, Chongqing University. His research interests include neural networks and their applications. He has authored over 100 journal and conference papers in these areas. He has over 20 research and applied grants.