Social Recommendation Using Euclidean Embedding

Wentao Li [*], Min Gao [†], Wenge Rong [‡], Junhao Wen [†], Qingyu Xiong [†], Ruixi Jia [†] and Tong Dou [†]
[*]Centre for Artificial Intelligence, School of Software, Faculty of Engineering & Information Technology, University of Technology Sydney
[†]School of Software Engineering, Chongqing University
[‡]School of Computer Science and Engineering, Beihang University
Emails: wentao.li@student.uts.edu.au, {gaomin, wjhcqu, xiong03, jiarx, doutong}@cqu.edu.cn, w.rong@buaa.edu.cn

***Abstract*—Traditional recommender systems assume that all the users are independent, and they usually face the cold start and data sparse problems. To alleviate these problems, social recommender systems use social relations as an additional input to improve recommendation accuracy. Social recommendation follows the intuition that people with social relationships share some kinds of preference towards items. Current social recommendation methods commonly apply the Matrix Factorization (MF) model to incorporate social information into the recommendation process. As an alternative model to MF, we propose a novel social recommendation approach based on Euclidean Embedding (SREE) in this paper. The idea is to embed users and items in a unified Euclidean space, where users are close to both their desired items and social friends. Experimental results conducted on two real-world data sets illustrate that our proposed approach outperforms the state-of-the-art methods in terms of recommendation accuracy.**

I. Introduction

Recommender systems have successfully been applied in various areas to reduce the information overload problem [1], [2]. For example, Amazon website provides personalized products to customs by analysis users' preference [3]. Recommender systems bring great values to both customers and producers, but traditional recommender systems assume users' preference towards items are independent [4]. Traditional recommender systems face the cold start and data sparse problems due to the limited ratings given by users [5].

With the rapid development of online social networks, social relations between users becomes easy to obtain [6], [7]. Based on the intuition that people with social relations show kinds of common preference [8], many researchers try to incorporate social information to improve recommendation results, which is called **social recommendation** in this paper [9]. Social recommender systems generate recommendations based on both rating information and social information, thus, they can solve the problems existing in traditional recommender systems by making use of the internal relationships between social friends [7], [9].

The core of social recommender systems is to incorporate social information into the recommendation process [9]. Many notable works have been done, and most of them are on the basic of traditional Collaborative Filtering (CF) methods by adding social factors to users [8], [9]. For model-based social recommendation methods, current research normally applies Matrix Factorization (MF) to make use of social relations. For example, Jamali et al. [10] combine the mechanism of trust propagation with matrix factorization to force users with social relations to gain similar latent preference. Ma et al. [11] design a matrix factorization objective function with the social regularization term to represent social constraints between users.

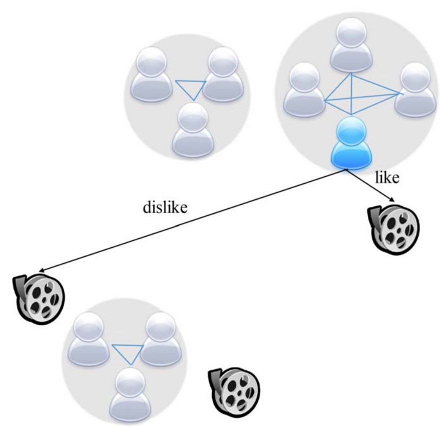


Fig. 1. An illustration of embedding users and items in a unified Euclidean space with social constrains

By exploiting the users' preference hidden in social networks, social recommender systems are useful even when ratings of a user are limited [12]. Existing social recommendation approaches using MF first obtain user latent vectors and item latent vectors by decomposing rating matrix with social relations, then predict missing values by multiple these two types of latent vectors [9], [13]. The whole process is like a black box, therefore, the mechanism behind was hard to understand because only latent factors of users and items can be obtained [13], [14].

To solve the above problems, in this paper, we propose a novel social recommendation model based on Euclidean embedding. We modify the matrix factorization model by using Euclidean embedding instead. We embed users and items

Corresponding author, Min Gao. E-mail: gaomin@cqu.edu.cn

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

in a common space. Moreover, we use social relations to make users with social relations be close with each other, thus to solve the data sparse problem when a users' rating information is limited. Also, the distance between a user and an item is inversely proportional to the rating. Thus, the recommendation results of our method are easy to understand by people. The main idea of our proposed method is shown in Fig. 1.

From Fig. 1, we can find that users and their desired items lie together, and users are near to their social friends, too. Here **social friends** of a user mean those people who are connected with this user in a social network. Formally, for user u, social friends of u are denoted by $N(u)$, which means neighbours of u in a social network. The Euclidean Embedding (EE) method was first proposed by Khoshneshin et al. [13]. However, in this approach, social relations are not introduced, and this method is likely to encounter the cold start and data sparse problems.

In this paper, we modified the EE model by adding social information to make users close to both their favorited items and social friends. The main contribution of this paper is to add social information into the EE approach. Our method solves the problems of EE approach, and the recommendation results of our method, which is the distance between users and items, are easier to understand by people. Experiment results on two real-life datasets show the effectiveness of our proposed method.

The rest of the paper is organized as follows: In the next section, we introduce related work about existing social recommendation methods and the EE model. We present the proposed approach in Section 3. Evaluation method and experimental results are presented in section 4 and section 5. We conclude in the last section.

II. Related Work

In this section, we first present some background knowledge about social recommendation. Then we give the Euclidean embedding model that is the basis of our proposed method.

A. Social Recommendation

The concept "social recommendation" used in this paper is defined as any recommendation with social relations as an extra input [9]. How to utilize social relations is the key of social recommendation methods [11]. Many current studies incorporate social information into the Matrix Factorization (MF) model, resulting in corresponding social recommendation models [10], [11], [15].

The MF model is used widely in traditional recommender systems [16], [17]. It projects users and items into a low-dimension space to find the latent patterns of rating [18], then predicts missing ratings using user latent vectors and items latent vectors. To train an MF model and get the latent vectors, it turns to minimize the objective function in Eq. 1.

$$\begin{aligned}\min_{p,q,b} \textstyle\sum_{i,j} I_{ij}[(r_{ij} - (\mu + b_i + b_j + p_i q_j'))^2 \\ +\lambda(\|p_i\|^2 + \left\|q_j\right\|^2 + b_i^2 + b_j^2)],\end{aligned} \tag{1}$$

where I_{ij} indicts whether user i has rated item j. $r_{ij}' = \mu + b_i + b_j + p_i q_j'$ is the predicted ratings, μ is the average of all ratings in a recommender system, b_i is the bias rating of user i from average, b_j is the bias rating of item j from average. p_i and q_j are the user latent factor vector and item factor vector. $\lambda(\|p_i\|^2 + \left\|q_j\right\|^2 + b_i^2 + b_j^2)$ is the term to avoid overfitting, λ is the algorithmic parameter to control overfitting. For simplicity, we make all terms to share the common overfitting parameter in this paper.

To introduce social information, many works have been done to inject social factors into the MF model. According to the way to integrate social relations into basic MF model, social recommendation methods can be classified into three types [9].

1) Regularization Methods: In this type, social relations are used to make users' latent factor vectors similar to those of their social friends. Specifically, these methods introduce social constrains into the regularization term. SocialMF [10] and Social Regularization [11] are two typical models in this type. Take SocialMF as an example, it introduces a new regularization term to force user latent vectors between social friends to be close with each other [10]. The objective function of SocialMF is listed in Eq. 2.

$$\begin{aligned}&\min_{u,v} \textstyle\sum_{ij} I_{ij}^R (r_{ij} - g(\alpha U_i V_j^T))^2 \\ &+\beta \sum_{i=1}^{n} (U_i - \textstyle\sum_{k \in N_i} S_{ik} U_k)^2 \\ &+\frac{\lambda}{2}(\|U\|^2 + \|V\|^2),\end{aligned} \tag{2}$$

where $R \in R^{m \times n}$ is a user-item rating matrix, $S \in R^{m \times m}$ is a user-user relationship matrix, $U_i \in R^{1 \times k}$ is a user latent factor vector, $V_j \in R^{k \times 1}$ is an item latent factor vector. m is the number of online users, n is the number of items, k is the dimension of latent factor vector, N_i is user i's social friend set.

$g(.)$ is the scale function, $|.|$ is the norm, α, β, λ are the penalty parameters. I_{ij}^R indicates whether user i gave ratings to item j. The meaning of this formula is that U_i is limited by user latent factor vectors of N_i.

2) Ensemble Methods: In this type, social relations are used to generate predicted ratings of a user to items. That is to use the combination of ratings from the target user and social friends to predict missing ratings. RSTE [15] is among this type of methods. The principle of RSTE is to generate users' rating to items by adding their own and their social friends' prediction values together [15]. The objective function of RSTE is listed in Eq. 3.

$$\begin{aligned}&\min_{u,v} \textstyle\sum_{ij} I_{ij}^R (r_{ij} - g(\alpha U_i V_j^T + (1-\alpha) \sum_{k \in N_i} S_{ik} U_k V_j^T))^2 \\ &+\lambda(\|U\|^2 + \|V\|^2),\end{aligned} \tag{3}$$

where N_i is user i's online friend set, U_i is user i's user latent factor vector, U_K is the user latent factor of user i's online friend. α is the parameter to decide the ratio of users' and their friends' ratings to the final rating to an item. The meaning of this is to use the mixture ratings as users' final ratings to items.

3) Co-factorization Methods: In this type, social relations are used to enable a user's preference to be the same in the rating space and social space. That is to say, instead of factorizing only user-item rating matrix, these methods co-factorizes both the user-item rating matrix and the user-user social matrix. Sorec [19] and LOCABAL [20] are two representative models in this type. For example, Sorec obtains users' latent vectors towards item by making this latent vectors in both rating matrix and social matrix [19]. The objective function of Sorec is listed in Eq. 4.

$$\begin{aligned} &\min_{u,v,z} \textstyle\sum_{ij} I_{ij}^{R}(r_{ij} - g(U_i V_j^T)) \\ &+\tfrac{\lambda_c}{2} \textstyle\sum_{ik} I_{ik}^{S}(S_{ik} - g(U_i X_k^T)) \\ &+\lambda(\|U\|^2 + \|V\|^2 + \|Z\|^2), \end{aligned} \tag{4}$$

where $Z_i \in R_{k\times 1}$ is a relationship latent factor vector. I_S^{ij} indicate whether user i has a social relationship with user j. λ is the penalty parameter. The meaning of this formula is that U_i is learned through factorizing both the rating matrix and the social matrix.

B. Euclidean Embedding Model

To make recommendation results more understandable for humans, Euclidean Embedding (EE) [13] is proposed as an alternation model to MF. This method embeds users and items into a unified Euclidean space, where the distance between users and items represent users' preference towards items [13], [14]. In the Euclidean space, short distance stands for strong preference.

More specifically, user u and item i are presented by their coordinates x_u and y_i, which is called **user position vector** and **item position vector** in this paper. The goal of EE is to make users near with their desired items. To train the EE model, it turns to solve the objective function in Eq. 5.

$$\begin{aligned} &\min_{x,y,b} \textstyle\sum_{i,j} I_{ij}[(r_{ij} - (\mu + b_i + b_j - \\ &(x_i - y_j)(x_i - y_j)^{'}))^2 + \lambda(\left\|x_i - y_j\right\|^2 + b_i^2 + b_j^2)], \end{aligned} \tag{5}$$

where $r_{ij}^{'} = \mu + b_i + b_j - (x_i - y_j)(x_i - y_j)^{'}$ is the predicted values of user i to item j. $(x_i - y_j)(x_i - y_j)^{'}$ is the Euclidean distance between x_i and y_j.

In the Euclidean space, users and items share the same representation [13], [14]. Since Euclidean embedding can be used for visualization, the results of the Euclidean embedding model are easy for people to understand [13]. Moreover, because items with high ratings are close to the target users, a local search can be used to retrieve top-*N* desired items for a target user, this leads to the fast recommendation [14].

However, the EE model only relies on rating information, therefore, it encounters the cold start and data sparsity problems faced by traditional recommender systems, too. Our method maintains the advantage of EE, while solving the problem of it. The main contribution of this paper is to incorporate social information into the EE model, which resulting in our proposed social recommendation model. To the best of our knowledge, social information has not been introduced in the EE model before. We hope that our proposed model benefits from both Euclidean embedding model and social information.

III. The Proposed Approach: SREE

In this section, we propose our **S**ocial **R**ecommendation model using **E**uclidean **E**mbedding, which is called **SREE**. SREE incorporates social information into the EE model, which tries to make users be near with both their favourite items and social friends.

SREE not only produces recommendation results understandable by people but also improves the recommendation accuracy of EE method. We first provide some preliminaries for SREE. Then, we present our approach (SREE) in detail. Finally is our parameter learning method.

A. Preliminaries

The purpose of many recommendation methods is to predict missing values of user-item rating matrix. Assume that we have m users, denoted by $U = (u_1, u_2, ...u_m)$, who give ratings to m items, denoted by $V = (v_1, v_2, ...v_n)$, in a recommender system. Let $R \in R^{m\times n}$ be the user-item rating matrix. If u_i gives a rating to v_j, then the value of r_{ij} is the rating, otherwise it is 0. The missing values, or the zeros, in r_{ij} is the target we want to predict.

Similarly, let $S \in R^{m\times m}$ be the user-user social matrix. If u_i has a connection with u_j in a social network, then S_{ij} is the strength and it means u_j is u_i's social friend, otherwise it is 0. Social recommendation assists the recommendation process by using the information in S to get better prediction accuracy.

We assume $x_i \in R^{d\times 1}$ to be the position vector of user i in a d-dimensional Euclidean space and $y_j \in R^{d\times 1}$ to be the position vector of item j. For SREE, given the rating matrix and social matrix, our purpose is to obtain x_i and y_j through known rating information and social information, and then make prediction based on x_i and y_j.

B. Model Definition

Recall that the main idea of EE is to find the user and item location in the Euclidean space based on existing ratings. As shown in Eq. 5, missing value of user i to item j is calculated by $r_{ij}^{'} = \mu + b_i + b_j - (x_i - y_j)(x_i - y_j)^{'}$. Where $(x_i - y_j)(x_i - y_j)^{'}$ is the Euclidean distance between x_i and y_j. The idea is that the shorter distance between user i and item j, the larger value of predicted value of r_{ij} will be. Thus EE can get the user and item position vectors according to Eq. 6 by the known rating values.

To incorporate social relations into the EE model, we first rewrite the prediction of a missing rating of EE. Since we want the positions of users to be near with those of their desired items and social friends, we need to add the distance between users and their friends into the previous Euclidean distance.

Let $(x_i - y_j)(x_i - y_j)^{'}$ denote the Euclidean distance between user i and item j, this term indicts that users will be near with their favourite items. Then, we get $(x_i - x_k)(x_i - x_k)^{'}$ as the distance between user i and i' social friend k, this term indicts that users will be close with their social friends. Because user i' social friend set is N_i, finally, we can get the prediction of the missing value given by user i to item j, which is shown in Eq. 6.

$$\begin{aligned} r'_{ij} = \mu + b_i + b_j - \alpha(x_i - y_j)(x_i - y_j)' \\ -\beta \textstyle\sum_{k \in N(i)} S_{ik}(x_i - x_k)(x_i - x_k)', \end{aligned} \tag{6}$$

where $N(i)$ is the set of social friends that u_i has been connected in a social network, S_{ik} indict the social strength between user i and user k. α is the **item parameter** which controls the effect of rating information. β is the **user parameter** which controls the effect of social information. α and β are parameters to make a tradeoff between these two kinds of information.

In this formula, social information is used to get the position vector of users and items. These position vectors show the preference of users towards items, and can be drawn in a common Euclidean space. Note, when users own ratings towards items are few, ratings from social friends can be assisted to learn position vector well. This is the reason why SREE can solve the data sparsity and cold start problem faced by EE.

To learn the user position vector x_i and item position vector y_j in the Euclidean space, we try to minimize the gap between the predicted ratings and known ratings. To avoid overfitting during the training process, we introduce the regularizing term. The overall objective function is shown in the Eq. 7.

$$\begin{aligned} \min_{x,y,b} \textstyle\sum_{i,j} I_{ij}[(r_{ui} - r'_{ui})^2 + \lambda(\left\|(x_i - y_j)\right\|^2 + \\ \textstyle\sum_{k \in N(i)} \|(x_i - x_k)\|^2 + b_i^2 + b_j^2)], \end{aligned} \tag{7}$$

where λ controls the strength of penalty, r'_{ui} is the predicted value of a rating given by Eq. 6.

When the process is over, user position vectors and item position vectors can be used to predict new ratings, thus finish the prediction precess. Moreover, note user position vectors and item position vectors lie in a unified space, therefore, it is easy to see why the prediction values are high or low. Moreover, a recommendation can be generated by calculating the distance between users and items, that means our method can be combined with the *KNN* method to get fast top-N recommendation by only querying the distance.

C. Parameter Learning

To get user and item position vector, we need to solve the Eq. 6. Here, we apply the gradient descent algorithm to learn parameters of SREE [21]. There are four parameter we need to learn: user bias b_i, item bias b_j, user position vector x_i, and item position vector y_j. The updating rules for each parameter is shown as follows:

$$b_i \leftarrow b_i + \eta(e_{ij} - \lambda b_i), \tag{8}$$

$$b_j \leftarrow b_j + \eta(e_{ij} - \lambda b_j), \tag{9}$$

Algorithm 1 The process of SREE

Input: Rating matrix R, Social matrix S, Dimension d, Traget user i, Target item j

Output: Predicted rating of r_{ij}

1) Compute predicted values of all existing ratings according to Eq. 6.
2) Get the objective function according to Eq. 7.
3) Solve the parameters of SREE according to Eqs. 8-11.
4) Find the user bias b_i and the user position vector x_i of user i
5) Find the item bias b_j and the item position vector y_j of item j
6) Return the predicted values of r_{ij} according to Eq. 6

TABLE I
DATASET STATISTICS

Dataset	# User	# Item	# Ratings	# Relations
FilmTrust	1508	2071	35497	1853
Ciao	7375	99749	278483	111781

$$x_i \leftarrow x_i - \eta[(x_i - y_j)(\alpha e_{ij} + \lambda) + \textstyle\sum_{k \in N(i)} (x_i - x_k)(\beta e_{ij} + \lambda)], \tag{10}$$

$$y_j \leftarrow y_j + \eta(x_i - y_j)(\alpha e_{ij} + \lambda), \tag{11}$$

where $e_{ij} = r_{ij} - r'_{ij}$ is the error between the genius rating and predict rating, η is learning rate, α and β are model parameters.

The learning process will continue until it reaches a fixed number of iteration. After that, user position vectors and item position vectors in Euclidean space can be obtained. These vectors can not only be used to predicted missing value but also be used to generate recommendation results by calculating the distance. The whole process of SREE is shown in Algorithm. 1.

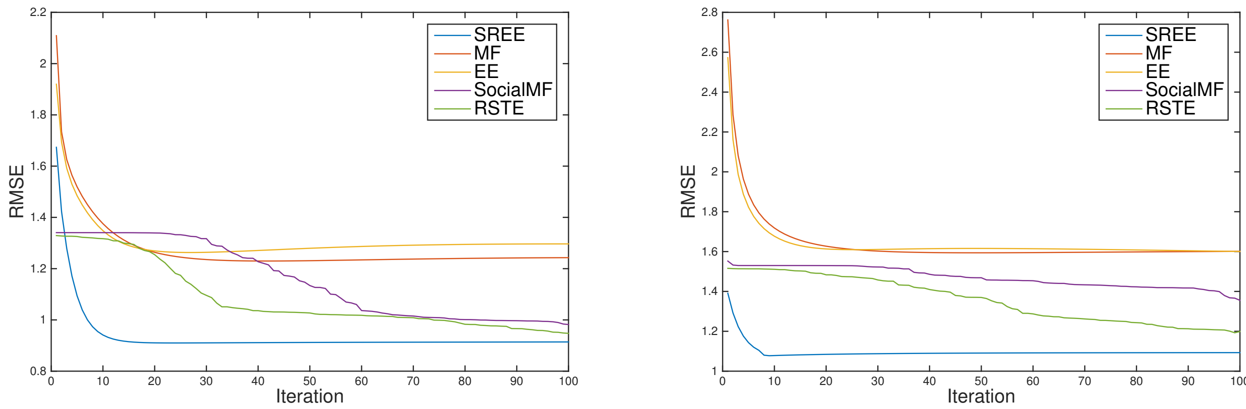
IV. EXPERIMENTAL EVALUATION

A. Datasets

We chose two real-world datasets for our experiments. FilmTrust is a dataset crawled from the entire FilmTrust website [22]. Ciao is a dataset crawled from ciao.co.uk website [23]. These two datasets include the rating information and social information, and are widely used in social recommendation area [9]. The social information in these two datasets is about trust. The statistics of these two datasets are shown in Table 1.

B. Evaluation Metric

We use the Root Mean Square Error (RMSE) to measure the performance of our proposed model in comparison with other related methods. RMSE is used to check the quality of recommendation methods for predicting missing ratings [24]. The lower RMSE is, the better a recommendation method is. RMSE is defined in Eq. 12.



(a) FilmTrust

(b) Ciao

Fig. 2. Comparison of performance between different methods on predicting missing ratings

$$RMSE = \sqrt{\frac{\sum_{i,j} (r_{ij} - r'_{ij})^2}{N}}, \tag{12}$$

where r_{ij} is the rating user i gives to item j, r'_{ij} is the rating predicted by a method, N is the number of tested ratings.

C. Baselines

To show the performance of our method, we compare SREE with the following methods.

- MF [16]: Matrix factorization is a typical method used in traditional recommender systems. It factorizes the user-item rating matrix to get user latent vectors and item latent vectors.
- EE [13]: Euclidean embedding is a substitution method of MF in traditional recommender systems. It embeds users and items in a common space and it uses the rating matrix to get user position vectors and item position vectors in the Euclidean space.
- SocialMF [10]: Social network matrix factorization introduces social information into the MF model and it forces user latent vectors to be similar to those of social friends by adding a regularization term.
- RSTE [15]: Recommend with social trust ensemble is another typical social recommendation model. RSTE predicts a missing rating by combining the predicted values of users and those of their social friends.

V. Experimental Results

We conducted two types of experiments. The first one is to compare between different recommendation methods about the ability to predict missing ratings. The second one is to see the effect of parameters on recommendation performance.

A. Performance for Predicting Missing Values

Predicting missing values is the main goal of many existing recommendation methods [25], [26]. This part we report results on predicting missing values in a test set. We use a 5-fold cross-validation strategy and take the average values as the final results. We implemented MF, EE and our proposed method SREE. We reused the implementations of SocialMF and RSTE in LibRec [27].

According to the settings given in [13], we set the learning rate $\eta = 0.05$ for all the methods. We fix the dimension of latent vector used in MF, socialMF, and RSTE to be 50, which is also adopted in [13]. We fix the dimension of position vector used in EE and SREE to be 50, too. For other parameters we use cross-validation method to get the optimal values. We run gradient descend algorithm 100 times and draw the values of RMSE in the test set as the gradient descent algorithm proceeds in Fig. 2.

Fig. 2 shows the performance of each recommendation methods on the FilmTrust dataset and Ciao dataset. From Fig. 2, we can find that social recommendation methods such as SREE, SoicalMF, and RSTE beat traditional recommendation methods on both two datasets. This justifies the value of social information in improving recommendation performance. SocialMF and RSTE show some fluctuates while others decrease as the number of iteration increases.

Moreover, as seen from Fig. 2, we find SREE outperforms all other methods on the two datasets in terms of prediction accuracy. RSTE has the best performance among the baselines and its performance nearly the same as our proposed model. SREE improves MF, EE, SocialMF, and RSTE by 41.85%, 35.98%, 9.8%, and 4.11% on the first dataset. SREE improves MF, EE, SocialMF, and RSTE by 41.48%, 41.42%, 25.24% and 5.4% on the second dataset. Among all five methods, MF ranks the lowest and SREE ranks the best. Therefore, we can draw the conclusion that the performance of SREE is better than state-of-the-art algorithms.

B. Parameter Effect Analysis

The proposed model includes three parameters to decide according to Eq. 4, i.e. item parameter α, user parameter β, and the dimension of position vector d. In this part, we discuss the effect of parameters on prediction accuracy. We show experiments conducted on the FilmTrust dataset to report results.

1) Parameter Effect Analysis of Dimension d: Dimension d is the parameter we took for the length of user position vector and item position vector. We present the effect of dimension d on predicting missing values in Fig. 3. We set other parameters according to cross-validation, and we make the iteration number fixed to 30. We change the dimension d among 10, 25, 50, 75, and 100 to conduct the experiments. The results can be found in Fig. 3.

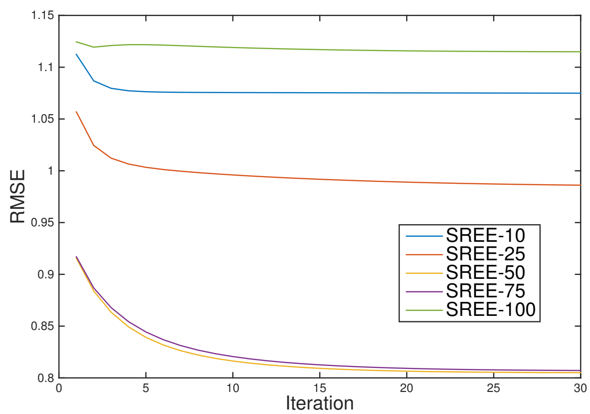


Fig. 3. Effect of dimension d on recommendation performance

As seen from Fig.3, it can be found that with the increase of dimension, RMSE first drops then increases. The main reason is that the dimension of user position vectors and item position vectors in the Euclidean space represent users and items in an implicit way. When the dimension is too low, the position vector can not express users and items well. When the dimension is too high, overfitting will appear. Therefore, a moderate value of d is important.

2) Parameter Effect Analysis of Item Parameter α: Item parameter α controls the effect of user own rating information. In the former section, we set α=0.01 and β=0.1 according to cross-validation. In this section, we fix the value of β to 0.1 and change the value of α among 0.01, 0.04, 0.1, and 0.5 to find the effect of item parameter α on recommendation accuracy, the results is shown in Fig. 4.

We fix the value of α to 0.01 and change the value β among 0.001, 0.005, 0.01, 0.05 to check the effect of user parameter β on recommendation accuracy, the result is shown in Fig. 5.

From Fig. 5 it can be found that the item parameter α has an important effect on the recommendation accuracy. As the values of α increase, RMSE first drops then increases. The reason behind is that α stands for the role of rating information in the recommendation. The higher α is, the larger effect of users' rating on the recommendation. When α is too small, we can not find anything useful about the user preference, so the RMSE is high. When α becomes too large, ratings hinder the effect of social information, and RSEE degenerates into the traditional recommendation and social factors become

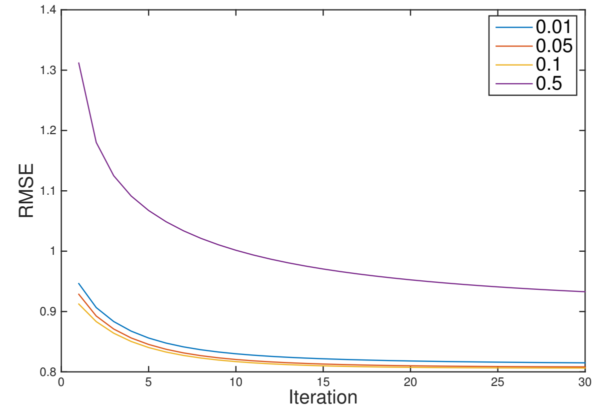


Fig. 4. Effect of item parameter α on recommendation performance

useless. The results become bad due to the limited number of ratings, which is the common problem faced by traditional recommender systems.

3) Parameter Effect Analysis of User Parameter β: User parameter β controls the effect of rating information provided by user's social friends. We fix the value of α to 0.01 and change the value β among 0.001, 0.005, 0.01, 0.05 to check the effect of user parameter β on recommendation accuracy, the result is shown in Fig. 5.

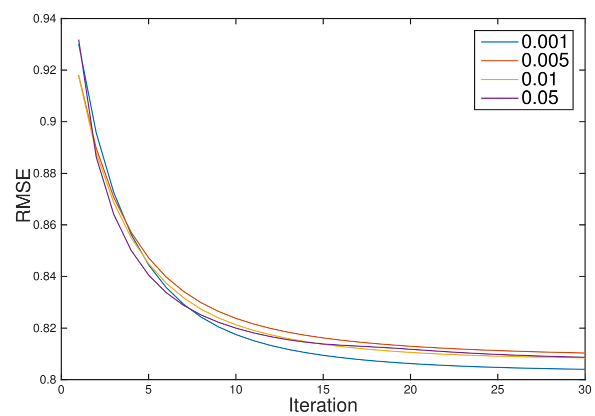


Fig. 5. Effect of user parameter β on recommendation performance

Fig. 5 shows that as the values of β increase, RMSE first increases slowly then drops slightly. User parameter β means the role of social information in the recommendation process. Note that β stands for the effect of ratings from social friends on the recommendation. As long as β is larger than a certain value, the recommendation accuracy is good due to the help of social factor. But when β becomes too large, rating information of a user are hindered, thus the recommendation accuracy drops. Though the proposed model depends on the choice of α and β, the good news is that we can use cross-validation to find the best setting for a practical problem.

VI. Conclusion

Social recommendation relieves the cold start and data sparsity problems faced by traditional recommendation. Therefore, in this paper, we concentrate on the problem of social recommendation and proposed a novel model based on Euclidean embedding. This approach makes use of both rating information and social information by embedding users and items in a shared space. Compared with current social recommendation models via matrix factorization, our model wins in terms of recommendation accuracy. Moreover, the recommendation results of our approach are easy to understand for humans because the distance between users and items represents users' preference towards items.

We assume users to be similar with their social friends, but this assumption may not always hold in reality. Therefore, in future, we plan to distinguish different types of social relations to improve our current model. Moreover, we use the gradient descent method to solve the proposed model, but this approach fails to work well in the face of large recommender systems. Thus, we plan to speed up the solution of our model with the approximation algorithm.

Acknowledgment

This research was supported by the Basic and Advanced Research Projects in Chongqing under Grant No. cstc2015jcyjA40049, the National Key Basic Research Program of China (973) under Grant No. 2013CB328903, the National Natural Science Foundation of China under Grant No. 61472021, the Fundamental Research Fund for the Central Universities under Grant No. 106112014CDJZR095502, and the China Scholarship Council.

References

[1] Z. Kang, C. Peng, and Q. Cheng, "Top-n recommender system via matrix completion," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[2] I. Elnabarawy, D. C. Wunsch, and A. M. Abdelbar, "Biclustering artmap collaborative filtering recommender system," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 2986–2991.

[3] Q. Zhao, "E-commerce product recommendation by personalized promotion and total surplus maximization," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 709–709.

[4] W. Luo, Z. Peng, and A. Deng, "Towards social recommendation based on probabilistic matrix factorization," *International Journal of u-and e-Service, Science and Technology*, vol. 9, no. 6, pp. 23–38, 2016.

[5] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2002, pp. 253–260.

[6] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, and B. Ling, "Lssl-ssd: Social spammer detection with laplacian score and semi-supervised learning," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2016, pp. 439–450.

[7] J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, and H. Liu, "Recommendation with social dimensions," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] Q. Zhang, J. Wu, H. Yang, W. Lu, G. Long, and C. Zhang, "Global and local influence-based social recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1917–1920.

[9] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[10] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 135–142.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296.

[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 transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[13] M. Khoshneshin and W. N. Street, "Collaborative filtering via euclidean embedding," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 87–94.

[14] Y. Wang, Y. Yu *et al.*, "Collaborative filtering via temporal euclidean embedding," in *Asia-Pacific Web Conference*. Springer, 2012, pp. 513–520.

[15] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 203–210.

[16] Y. Koren, R. Bell, C. Volinsky *et al.*, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[17] S. Kalloori, F. Ricci, and M. Tkalcic, "Pairwise preferences based matrix factorization and nearest neighbor recommendation techniques," in *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 2016, pp. 143–146.

[18] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Investigation of various matrix factorization methods for large recommender systems," in *2008 IEEE International Conference on Data Mining Workshops*. IEEE, 2008, pp. 553–562.

[19] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 931–940.

[20] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation." in *IJCAI*, 2013, pp. 264–269.

[21] H.-F. Yu, C.-J. Hsieh, S. Si, and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems," in *2012 IEEE 12th International Conference on Data Mining*. IEEE, 2012, pp. 765–774.

[22] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems." in *IJCAI*, 2013.

[23] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Etaf: An extended trust antecedents framework for trust prediction," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 540–547.

[24] A. Said and A. Bellogín, "Comparative recommender system evaluation: benchmarking recommendation frameworks," in *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 2014, pp. 129–136.

[25] X. Feng, A. Sharma, J. Srivastava, S. Wu, and Z. Tang, "Social network regularized sparse linear model for top-n recommendation," *Engineering Applications of Artificial Intelligence*, vol. 51, pp. 5–15, 2016.

[26] C. Park, D. Kim, J. Oh, and H. Yu, "Trecso: Enhancing top-k recommendation with social information," in *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 89–90.

[27] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems." in *UMAP Workshops*, 2015.