Social Collaborative Filtering Ensemble

Honglei Zhang[1], Gangdu Liu[2] *, and Jun Wu[1] **

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
[2] School of Civil and Transportation Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
{16120453, 15398518, wuj}@bjtu.edu.cn

Abstract. Collaborative filtering (CF) technique plays an important role in generating personalized recommendations, but its performance is challenged by the problems of data sparsity and cold start. Besides, different CF methods have their own advantages, so another tough issue is how to exploit the complementary properties of different methods. In this paper, we propose a general framework to ensemble three popularly used CF methods, termed as TriCF, aiming to further elevate recommendation accuracy. In order to alleviate the data sparsity problem, we incorporate social information into TriCF by graph embedding, denoted as SoTriCF. In particular, a mapping from social domain to rating domain is built by a neural network model, which can enhance the cold-start users' latent representation learned from rating data. Extensive experiments on three real-world datasets show that the proposed approaches achieve significant improvements over state-of-the-art methods.

Keywords: Recommender systems · Collaborative filtering · Social network · Latent representation.

1 Introduction

Recommender systems, which are capable of reaching the equilibrium between information producers and consumers, are now ubiquitous online. Collaborative filtering (CF) is the most successful and widely used recommendation technique, and numerous CF algorithms emerged in last two decades. Roughly speaking, such algorithms can be grouped into two categories, neighborhood methods and latent factor models. Neighborhood-based methods focus on finding the nearest neighbors of the active user [21] or target item [23], and make prediction based on the ratings from these neighbors. Latent factor models [12] primarily center on embedding both users and items into a common latent subspace, and then the prediction can be estimated by inner products between user and item latent

* This work was performed when the second author was a joint-training student of Beijing University of Civil Engineering and Architecture (BUCEA) and Beijing Jiaotong University (BJTU) from 2015 to 2018.

** Corresponding author.

factors. As demonstrated by [2], however, none of them is always dominant in various recommendation scenarios, and the performance of a CF algorithm is related to various factors, such as the number of users/items, and the density level of the user-item rating matrix. Hence, for further improve recommender systems, we need to integrate the merits of different CF approaches.

It is well-known that CF methods have to confront the problems of data sparsity and cold start. In recent years, in addition to the rating data, a large number of recommendation scenarios have emerged in which various additional information sources are available [24], e.g. the social relationships among users [25]. A few studies have demonstrated that incorporating such social information into CF algorithms is helpful to elevate the recommendation accuracy and to alleviate the problems of data sparsity and cold start [16, 15, 10, 17, 8]. However, such improvement is limited, because existing social recommender systems ignore a phenomenon that the social data is also very sparse, even though it is complementary to the rating data. In other words, dealing with the social sparsity is as important as the rating sparsity to establish a high-performance social recommender system.

In this paper, we intend to solve the problems mentioned above, by proposing the SoTriCF approach that integrates the merits of three popularly used CF methods to improve the recommendation accuracy. Moreover, SoTriCF encodes the social connections among users into low-dimensional and compact vectors, and takes that as side-information to enhance the CF ensembling, such that the sparsity of rating data and social data are jointly addressed. Specifically, we design a method to latent representation mapping based on a neural network model, which can enhance the latent representation of cold-start users learned from rating data using their social latent representations. The experimental analysis on three datasets shows that our SoTriCF approach outperforms other state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed SoTriCF approach. Section 4 reports on the experiments. Finally, Section 5 concludes this paper and raises several problems for future works.

2 Related Work

In this section, we briefly classify related work into three dimensions, i.e. collaborative filtering, social recommendation, and graph embedding.

Collaborative filtering (CF) is a family of algorithms widely used in recommendation systems. Neighborhood-based methods and latent factor models, can be differentiated depending on how the rating data are processed. Neighborhood-based methods use a certain similarity measure to select users (or items) that are similar to the active user (or the target item). Then, the prediction is calculated from the ratings of these neighbors. It can be further divided into user-based and item-based on the basis of whether the process of seeking neighbors is focused on similar users [21] or items [23]. Latent factor models aim at factorizing the user-item rating matrix, and use the factorized user-specific and item-specific matrices to make further missing data prediction [18, 11, 19, 12]. Different CF methods have their own advantages and are applicable to different scenarios, so it is natural to ensemble them for better prediction performance. Koren [11] presented an integrated model that combines latent factor model and item-based CF, however, which is very memory-consuming. In this paper, we try to ensemble three widely-used CF methods, i.e. the user-based CF, item-based CF, and matrix factorization (MF), through a unified formulation in a more efficient way.

Due to the potential value of social relations in recommender systems, social recommendation has attracted increasing attention in recent years. Existing social recommendation methods can be roughly categorized into three types [25], i.e., Regularization methods, Ensemble methods and Co-factorization methods. Regularization methods focus on a users's preference should be similar to that of their social friends [10, 17]. The basic idea of Ensemble methods is that the predicted rating for a given user is a combination of ratings from the user and their social network [15]. In Co-factorization methods, the underlying assumption is that the user should share the same user preference in the rating space and the social space [8]. However, similar to the rating data, the social data is also very sparse, which is one of challenges we tackle in this paper.

Furthermore, our approach is also related to graph embedding technique, which embeds complex social relations into a low-dimensional vector space. Such low-dimensional representation is much more denser than the traditional graph representation, so graph embedding is a potential solution to alleviating the sparsity of social network. Existing graph embedding methods can be broadly divided into three types, factorization based methods [22, 3], random walk based methods [20, 5] and deep learning based methods [26]. For comprehensive review of graph embedding, please refer to [4]. Recently, Zhang et al. [30] first extracted a implicit social network based on the affinity calculated from the user-item rating matrix, and then incorporated such social information into matrix factorization by graph embedding. However, an early empirical study [14] had demonstrated that the explicit social connection is more effective than the implicit social relationship inferred from rating data to enhance the recommendation accuracy. Hence the improvement made by [30] is limited. Differently, we try to exploit the explicit social connections by graph embedding to enhance the CF ensembling, aiming at addressing the sparsity of rating data and social data jointly.

3 The Proposed SoTriCF Approach

In this section, we will elaborate our SoTriCF approach whose framework is shown in Fig. 1. It consists of three main modules. The first module is a latent factor model that incorporates the merits of three popularly-used CF methods into a unified formulation, and is learned based on both rating data and social data. The second module is a graph embedding model which embeds social network into a low-dimensional vector space, in order to alleviate the sparsity

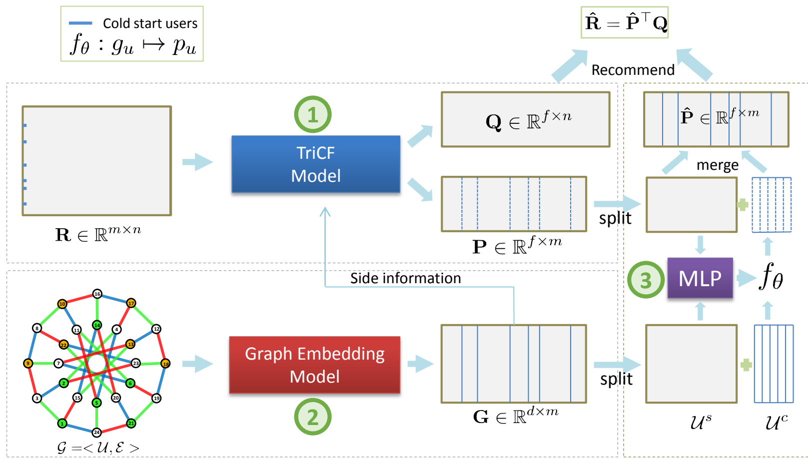


Fig. 1. Framework of our SoTriCF approach.

of social data. The last one is a Multi-Layer Perceptron (MLP) model that is to build a nonlinear mapping from the users social latent representation to rating latent representation (detailed in Section 3.4). Using such mapping, the cold-start users identified from the rating data can be enhanced by their social information.

3.1 Preliminaries

Let $\mathcal{U} = \{u_1, \cdots, u_m\}$ be a set with m users, and $\mathcal{I} = \{i_1, \cdots, i_n\}$ denote a set with n items. All interactions associating $\mathcal{U}$ with $\mathcal{I}$ can be represented by a user-item rating matrix $\mathbf{R} \in \{1, 2, 3, 4, 5\}^{m\times n}$ whose element r_{ui} indicates the rating of user u scored on item i. Furthermore, we define $\mathcal{I}_u \in \mathcal{I}$ be the set of items rated by a certain user u, and $\mathcal{U}_i \in \mathcal{U}$ denotes the set of users who have rated a certain item i. We also define $\mathcal{G} =< \mathcal{U}, \mathcal{E} >$ as the social network among users, where $\mathcal{E} = \{e_{uv}\}$ is a set of edge recording the friend relationship between users, and $\mathcal{F}_u \in \mathcal{U}$ denotes the set of friends of user u.

An efficient and effective solution to recommender systems is the MF models that factorize the user-item rating matrix, and utilize the factorized user and item vectors to predict missing data. Let $\mathbf{p}_u$ be the user-specific latent vector and $\mathbf{q}_i$ be the item-specific latent vector. For better modeling the user-item interaction, we take the biases of users and items into consideration, where μ denotes the overall average rating, b_u and b_i indicate the deviations of user u and item i, respectively. A typical cost function can be defined as

$$\begin{aligned} \min_{\mathbf{p}_u, \mathbf{q}_i, b_u, b_i} \frac{1}{2} \textstyle\sum_{(u,i)\in\Omega} [\, (r_{ui} - \mu - b_u - b_i - \mathbf{p}_u^T \mathbf{q}_i)^2 \\ + \quad \gamma(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2)\,] \end{aligned} \tag{1}$$

where Ω is a set of the index of observed entries, and $\gamma > 0$ is a free parameter. The first term of Eq. (1) is a fitting term that ensures the learned $\hat{\mathbf{r}}_{ui}$ to be consistent with the observed ratings, and the second term is the regularizer that is to alleviate model overfitting.

Another promising solution is neighborhood-based methods which use similarity measures to identify a neighborhood of the active user (or target item), and then calculate the prediction using the ratings from these neighbor users (or items). For example, the user-based method is to predict the unobserved rating r_{ui} by averaging the observed ratings from the neighbors of the user u

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_u \wedge i \in \mathcal{I}_v} S_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_u} S_{uv}} \tag{2}$$

where $\mathcal{N}_u$ denotes the user u's neighbors, and S_{uv} is the similarity between user u and v, which can be calculated by Pearson Correlation Coefficient (PCC) [1]

$$S_{uv} = \frac{\sum_{i \in \mathcal{I}_u \cap \mathcal{I}_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_u \cap \mathcal{I}_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in \mathcal{I}_u \cap \mathcal{I}_v} (r_{vi} - \bar{r}_v)^2}}. \tag{3}$$

Likewise, the item-based method is to predict the unobserved rating r_{ui} by averaging the observed ratings from the neighbors of the item i

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{N}_i \cap \mathcal{I}_u} S_{ij} r_{uj}}{\sum_{j \in \mathcal{N}_i \cap \mathcal{I}_u} S_{ij}} \tag{4}$$

where $\mathcal{N}_i$ denotes a set of item i's neighbors, and S_{ij} is the similarity between item i and j, which can be also calculated by PCC

$$S_{ij} = \frac{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in \mathcal{U}_i \cap \mathcal{U}_j} (r_{uj} - \bar{r}_u)^2}}. \tag{5}$$

An empirical study demonstrates that none of CF methods is dominant in all recommendation tasks, even though MF models achieves slightly better performance in most cases. Considering this, in the next subsection, we will ensemble the three CF methods mentioned above for higher prediction accuracy.

3.2 Ensembling Three CF Methods

A 'good' CF solution should maintain the user-user affinity and the item-item affinity when it model the users' preference in a common subspace of users and items, such that the data geometry in the original space could be preserved as much as possible. Based on this intuition, we propose the TriCF method by extending the regular MF formulation Eq. (1) by adding the smoothness terms of users and items

$$\min_{\mathbf{p}_u, \mathbf{q}_i, b_u, b_i} \frac{1}{2} \sum_{(u,i) \in \Omega} [\ (r_{ui} - \mu - b_u - b_i - \mathbf{p}_u^T \mathbf{q}_i)^2$$

$$
\begin{aligned}
+ \quad & \alpha \sum_{v \in \mathcal{N}_u} S_{uv} \|\mathbf{p}_u - \mathbf{p}_v\|^2 \\
+ \quad & \beta \sum_{j \in \mathcal{N}_i} S_{ij} \|\mathbf{q}_i - \mathbf{q}_j\|^2 \\
+ \quad & \gamma(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) \ ]
\end{aligned} \tag{6}
$$

where $\alpha > 0$ and $\beta > 0$. The second and third terms of Eq. (6) are user smoothness and item smoothness respectively, and S_{uv} and S_{ij} can be calculated by Eq. (3) and Eq. (5) respectively.

It is noting that the two smoothness terms in Eq. (6) follow the spirit of the user-based method (Eq. (2)) and the item-based method (Eq. (4)). This is motivated by the assumption that if two users impose similar ratings on some items, they will have similar rating behaviors on the remaining items. Likewise, if two items are rated similarly by a portion of the users, they will still be rated similarly by the remaining users. By adding the user and item smoothness terms into the regular MF formulation, our TriCF can collaboratively exploit the merits of both latent factor models and neighborhood-based methods.

It is well-known that the performance of CF methods is often limited by the sparsity inherent in the rating data, and thus we will study the solution to incorporating social signal into TriCF formulation in the next subsection.

3.3 Exploiting Social Signal in TriCF

To exploit the social signal, we further extend the TriCF formulation by introducing a social smoothness term. As a result, we further change Eq. (6) to

$$
\begin{aligned}
\min_{\mathbf{p}_u, \mathbf{q}_i, b_u, b_i} \mathcal{L} = \frac{1}{2} \textstyle\sum_{(u,i) \in \Omega} & \{ \ (r_{ui} - \mu - b_u - b_i - \mathbf{p}_u^T \mathbf{q}_i)^2 \\
+ \quad & \alpha \Big[\lambda \sum_{v \in \mathcal{N}_u} S_{uv} \|\mathbf{p}_u - \mathbf{p}_v\|^2 + (1-\lambda) \sum_{v \in \mathcal{F}_u} S_{uv}^{\mathcal{N}} \|\mathbf{p}_u - \mathbf{p}_v\|^2 \Big] \\
+ \quad & \beta \sum_{j \in \mathcal{N}_i} S_{ij} \|\mathbf{q}_i - \mathbf{q}_j\|^2 \\
+ \quad & \gamma(\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2 + b_u^2 + b_i^2) \ \}
\end{aligned} \tag{7}
$$

where $\lambda > 0$, and $S_{uv}^{\mathcal{N}}$ is the social similarity estimated from social connections. Now we have two user smoothness terms, which are from rating data and social data respectively.

A local minimum of the objective function given by Eq. (7) can be found by performing gradient descent or statistic gradient descent in variables $\mathbf{p}_u$, $\mathbf{q}_i$, b_u and b_i,

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{p}_u} = & \sum_{i \in I_u} e_{ui} \cdot \mathbf{q}_i + \alpha(\lambda \sum_{v \in \mathcal{N}_u} S_{uv}(\mathbf{p}_u - \mathbf{p}_v) \\
& + (1-\lambda) \sum_{v \in \mathcal{F}_u} S_{uv}^{\mathcal{N}}(\mathbf{p}_u - \mathbf{p}_v)) + \gamma \mathbf{p}_u
\end{aligned} \tag{8}
$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = \sum_{u \in U_i} e_{ui} \cdot \mathbf{p}_u + \beta \sum_{j \in \mathcal{N}_i} S_{ij}(\mathbf{q}_i - \mathbf{q}_j) + \gamma \mathbf{q}_i \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial b_u} = \sum_{i \in I_u} e_{ui} + \gamma b_u \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \sum_{u \in U_i} e_{ui} + \gamma b_i \tag{11}$$

where $e_{ui} = \hat{r}_{ui} - r_{ui}$ indicates the prediction error for user u on item i.

Given the social network, we can compute $S_{uv}^{\mathcal{N}}$ based on the network structure, as done in the link prediction study [13]. However, social data is also very sparse. To this end, we advocate to use graph embedding technique, which can embed social network into a low-dimensional and compact vector space. In such space, each user node is represented by a low-dimensional vector, and thus the sparsity problem can be alleviated to some degree.

Let $\mathbf{g}_u \in \mathbb{R}^d$ and $\mathbf{g}_v \in \mathbb{R}^d$ be two latent representation learned by DeepWalk, more details of which can be found in [20]. Given $\mathbf{g}_u$ and $\mathbf{g}_v$, we can calculate $S_{uv}^{\mathcal{N}}$ by a Gaussian kernel

$$S_{uv}^{\mathcal{N}} = \exp\left(-\frac{\|\mathbf{g}_u - \mathbf{g}_v\|^2}{\sigma^2}\right) \tag{12}$$

where σ is a scale parameter that can be tuned by a local scaling technique, the effectiveness of which has been verified by spectrum clustering [29] and manifold ranking [28].

As mentioned before, the problem of cold-start users is one of challenges for CF technique, and it is desirable to improve the experience of the cold-start users who rate very few items. Hence we will study the solution to enriching cold-start users using social signal in the next subsection.

3.4 Enhancing the Latent Representation of Cold-start Users

For any user u, we can learn two latent representations $\mathbf{p}_u \in \mathbb{R}^f$ and $\mathbf{g}_u \in \mathbb{R}^d$ from rating data and social data respectively. We refer to the rating domain as the target domain and the social domain as the source domain. Our solution to enhancing cold-start users is based on an intuition that there is a potential relation between the two domains, and we aim to build a mapping to capture this potential relation. With such mapping, we can improve the representation of a cold-start user learned from the target domain using its another representation learned from the source domain. As demonstrated by [9], any continuous mapping can be realized by a neural network with one hidden layer, which can capture the more complex relationship than linear transformation. So we apply a multi-layer perceptron (MLP) model to construct the mapping from the source domain to the target domain.

In order to guarantee its robustness to noise caused by data sparsity in rating data and social data, only the entities with sufficient data in both domains are used to learn the mapping function. As is common practice, we define users that rated less than five items as cold-start users in rating domain and users that have less than four friends as cold-start users in social domain. Let $\mathcal{U}_{cs}$ and $\mathcal{U}_{ct}$ denote the sets of cold-start users identified from the source domain and the target domain respectively. It worth noting that the users in $\mathcal{U}_{cs} \cap \mathcal{U}_{ct}$ are not useful to representation mapping, because their $\mathbf{p}_u$s and $\mathbf{g}_u$s are unreliable. Fortunately, in most datasets used for social recommendation we observed that very few users are in $\mathcal{U}_{cs} \cap \mathcal{U}_{ct}$. We define $\mathcal{U}_{train} = (\mathcal{U} \setminus \mathcal{U}_{cs}) \cap (\mathcal{U} \setminus \mathcal{U}_{ct})$ as the training user set and $\mathcal{U}_{test} = \{u | u \notin \mathcal{U}_{cs} \wedge u \in \mathcal{U}_{ct}\}$ as the testing user set. The users in $\mathcal{U}_{train}$ are not 'cold-start' in both domains and used to learn the mapping function, while the users in $\mathcal{U}_{test}$ are not 'cold-start' in the source domain yet 'cold-start' in the target domain because we only pay attention on the target domain.

Given the training user set $\{< \mathbf{g}_u, \mathbf{p}_u >, u \in \mathcal{U}_{train}\}$, we can learn a MLP model $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}^f$, where $\mathbf{g}_u$ is the input representation (feature) and $\mathbf{p}_u$ is the output representation (label). Based on such mapping function, for a user $v \in \mathcal{U}_{test}$, we can take its reliable social representation $\mathbf{g}_v$ as input, and obtain its improved rating representation by $\hat{\mathbf{p}}_v = f_\theta(\mathbf{g}_v)$. At last, the recommendation on the cold-start users is made using their improved rating representations.

4 Experimental Analysis

In this section, we first introduce some necessary preparations for experiments, and then present the experimental results of our approach compared with several other state-of-the-art recommendation methods.

4.1 Experimental setup

We employ three datasets (FilmTrust, Douban and Ciao) which are publicly available on the web to make our experiments reproducible. FilmTrust is a small-scale dataset [7]. In this dataset, 95% of users rated no more than 50 items, in which 15% of users are cold-start users. Moreover, the social data is sparse as well, and the average number of friends per user is only 1.22. Douban is a medium-scale dataset [17]. In this dataset, the number of items is far more than that of users and each user has 12.07 friends in average. Ciao is a large-scale dataset [6]. In such dataset, 90% of users rated less than 50 items, in which 8% of users are cold-start users. The social density of this dataset is only 0.04% around and the average amount of each users friends is 15.16. A more detailed statistics about these three datasets are summarized in Table 1 and Table 2, in terms of rating information and social information respectively.

We utilize two metrics to measure the prediction accuracy, root mean square error(RMSE) and mean absolute error(MAE), defined as

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in\mathcal{T}}(r_{ui} - \hat{r}_{ui})^2}{|\mathcal{T}|}},\ MAE = \frac{\sum_{(u,i)\in\mathcal{T}} |r_{ui} - \hat{r}_{ui}|}{|\mathcal{T}|}$$

Table 1. Statistics of three datasets about ratings data.

Datasets	Users	Items	Ratings	Avg.Num.Ratings per User	Avg.Num.Ratings per Item	Density
FilmTrust	1,508	2,071	35,497	23.54	17.14	1.136%
Douban	2,963	39,694	894,887	301.9	22.54	0.760%
Ciao	7,375	105,114	284,086	38.52	2.66	0.037%

Table 2. Statistics of three datasets about social data.

Datasets	Followers	Followees	Links	Avg.Num.Friends per User	Density
FilmTrust	609	732	1,853	1.23	0.415%
Douban	2,745	2,741	35,771	12.07	0.475%
Ciao	6,792	7,297	111,781	15.16	0.225%

where $\mathcal{T}$ denotes the testing set and $|\bullet|$ is the number of a set. The smaller RMSE and MAE values are, the better prediction performance is.

We compare the proposed TriCF and SoTriCF approaches with a number of related existing methods. The compared methods are enumerated as follows, where the first two methods are used in the traditional recommendation scenarios, and the last five methods are designed for the social recommendation tasks.

- **PMF**: A canonical latent factor model which factorizes both users and items in a common subspace [18].
- **SVD++**: A latent factor model which models the biases of users and items, and incorporates the user implicit feedback into matrix factorization formulation [11].
- **SoRec**: A matrix factorization model with social regularization [16].
- **RSTE**: An ensemble method which combines a basic matrix factorization model with a social-based neighbor model [15].
- **SocialMF**: A social model which forces on the preference of a user to be closer to the average of their social friends [10].
- **SoReg**: A social model which forces on the preference closeness of two connected users in social network controlled by similarity based on ratings data [17].
- **TrustSVD**: A extended method of SVD++, which incorporates both rating implicit feedback and social implicit feedback into formulation [8].

For all the compared methods, we adopt the parameter configuration as described in previous works or by experimental selection for the best results. The parameters are tuned via 5-fold cross validation and grid search technique, and for our approach, we set the parameters $\lambda = 0.3$ for FilmTrust and Ciao, $\lambda = 0.5$ for Douban; and $\alpha = 0.002$, $\beta = 0.001$, $\gamma = 0.05$ for the three datasets. We set $f = 10$ to learn the latent representation of users and items, and $d = 30$ to learn the node representation in graph embedding.

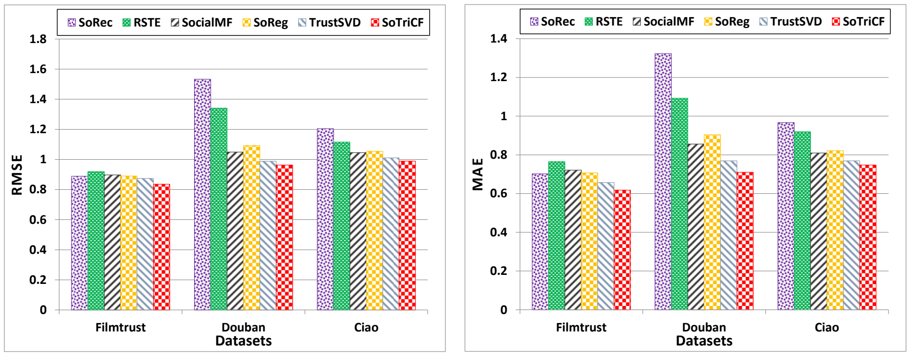
Table 3. Performance of our SoTriCF approaches with several existing methods.

Datasets	**Filmtrust**		**Douban**		**Ciao**	
Metrics	RMSE Improve	MAE Improve	RMSE Improve	MAE Improve	RMSE Improve	MAE Improve
PMF	0.87869 7.99%	0.66427 6.21%	0.78625 2.91%	0.62643 3.10%	1.02434 4.83%	0.78124 3.84%
SVD++	0.86800 6.85%	0.64954 4.09%	0.84862 10.04%	0.65794 7.74%	1.03408 5.72%	0.76056 4.15%
TriCF	**0.83321** 2.96%	**0.64779** 3.83%	**0.77969** 2.09%	**0.62257** 2.50%	**0.99524** 2.04%	**0.75676** 3.67%
SoRec	0.83923 3.66%	0.64388 3.24%	0.90601 15.74%	0.69780 13.01%	1.13637 14.21%	0.83797 13.00%
RSTE	0.84654 4.49%	0.65970 5.56%	0.81357 6.17%	0.63896 5.00%	1.00414 2.91%	0.76473 4.67%
SocialMF	0.84782 4.64%	0.65727 5.21%	0.79398 3.85%	0.62561 2.97%	0.99508 2.03%	0.74941 2.72%
SoReg	0.83114 2.72%	0.64120 2.84%	0.78006 2.14%	0.61729 1.67%	0.99506 2.03%	0.75176 3.02%
TrustSVD	0.84404 4.21%	0.65178 4.42%	0.79347 3.79%	0.62555 2.96%	1.03295 5.62%	0.75903 3.95%
SoTriCF	**0.80851**	**0.62300**	**0.76340**	**0.60701**	**0.97490**	**0.72902**

4.2 Result Analysis

At first, we compare performance of our TriCF and SoTriCF approaches with several existing methods. The corresponding quantitative results are shown in Table 3, and column 'Improve' indicates the relative improvements that our TriCF and SoTriCF achieve relative to their basic models respectively. From experimental results, the following interesting observations are revealed.

- By comparing the three CF methods for regular recommendation, our TriCF approach outperforms the two baselines PMF and SVD++, which verfiies the benefit of ensembling latent factor models and neighborhood methods. It is worth noting that the performance of SVD++ is not as excellent as that reported in [11], because the implicit feedback signal is not available in the three datasets used in our experiments.
- In most cases, the performance of social CF methods is better than the regular CF methods. This observation reveals that the social signal is an effective compensation to the rating signal, which is consistent with previous studies [16, 15, 10, 17, 8].



(a) RMSE (b) MAE

Fig. 2. The performance comparisons for social recommendation on cold-start users.

- In comparison of the six social CF methods, our SoTriCF achieves the best performance on all datasets. The main difference between our SoTriCF approach and other five social CF methods is that SoTriCF exploits the social signal in a low-dimension embedding space of social network, while the other five methods exploit the social signal directly. It well demonstrates that adopting graph embedding is beneficial to alleviate the sparsity of original social signal. That is, the motivation of this work is empirically verified.

Furthermore, we investigate the performance of various social CF methods on cold-start users to verify whether our representation mapping solution to cold-start users is effective or not. Experimental results of RMSE and MAE for social recommendation task on cold-start users are presented in Fig. 2(a) and Fig. 2(b). From the results, we can observe that the RMSE and MAE values generated by our method are much smaller than those generated by other comparison methods in all datasets. This confrims that our approach can consistently generate better results than the state-of-the-art social recommendation methods in the case of cold start.

5 CONCLUSIONS

In this work, we have proposed a SoTriCF approach that ensembles three popularly used CF methods within a unified learning formulation. To well handle the sparsity inherent in both rating data and social data, we exploit the social information in CF ensemble by graph embedding. In order to alleviate the cold start problem, we also developed a solution to latent representation mapping from social domain and rating domian based on a neural network model, which can enhance the cold-start users latent representation learned from rating data using their social information. Experimental study has validated the superiority of the proposed approaches in comparison to state-of-the-art methods.

In the future, we will extend our SoTriCF approach by incorporating more side-information, in order to further improve its prediction performance. Moreover, inspired by the evolving graph mining technique [27], another extension of this work is to develop the online version of SoTriCF to take the social data stream into consideration.

Acknowledgments. The authors would like to thank the anonymous reviewers for their constructive suggestions. This work was supported in part by the 'Natural Science Foundation of China' #61671048 and the 'Research Foundation for Talents of Beijing Jiaotong University' #2015RC008. The corresponding author of this paper is Jun Wu (e-mail: wuj@bjtu.edu.cn).

References

1. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI. pp. 43–52 (1998)
2. Cacheda, F., Carneiro, V., Fernández, D., Formoso, V.: Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans. Web **5**(1), 2 (2011)
3. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: CIKM. pp. 891–900 (2015)
4. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowl.-Based Syst. (2017)
5. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD. pp. 855–864 (2016)
6. Guo, G., Zhang, J., Thalmann, D., Yorke-Smith, N.: Etaf: An extended trust antecedents framework for trust prediction. In: ASONAM. pp. 540–547 (2014)
7. Guo, G., Zhang, J., Yorke-Smith, N.: A novel bayesian similarity measure for recommender systems. In: IJCAI. pp. 2619–2625 (2013)
8. Guo, G., Zhang, J., Yorke-Smith, N.: Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI. pp. 123–129 (2015)
9. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural networks **2**(5), 359–366 (1989)
10. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys. pp. 135–142 (2010)
11. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: SIGKDD. pp. 426–434 (2008)
12. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
13. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A Stat. Mech. Appl. **390**(6), 1150–1170 (2011)
14. Ma, H.: An experimental study on implicit social recommendation. In: SIGIR. pp. 73–82 (2013)
15. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: SIGIR. pp. 203–210 (2009)
16. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM. pp. 931–940 (2008)

17. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM. pp. 287–296 (2011)
18. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: NIPS. pp. 1257–1264 (2008)
19. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: SIGKDD. vol. 2007, pp. 5–8 (2007)
20. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD. pp. 701–710 (2014)
21. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW. pp. 175–186 (1994)
22. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
23. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW. pp. 285–295 (2001)
24. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACM. Comput. Surv. **47**(1), 3 (2014)
25. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Soc. Netw. Anal. Min. **3**(4), 1113–1133 (2013)
26. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: SIGKDD. pp. 1225–1234 (2016)
27. Wang, H., Zhang, P., Zhu, X., Tsang, I., Chen, L., Zhang, C., Wu, X.: Incremental subgraph feature selection for graph classification. IEEE Trans. Knowl. Data Eng. **29**(1), 128–142 (2017)
28. Wu, J., Li, Y., Feng, S., Shen, H.: A self-immunizing manifold ranking for image retrieval. In: PAKDD. pp. 426–436 (2013)
29. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS. pp. 1601–1608 (2005)
30. Zhang, C., Yu, L., Wang, Y., Shah, C., Zhang, X.: Collaborative user network embedding for social recommender systems. In: SDM. pp. 381–389 (2017)