Northumbria Research Link

www.northumbria.ac.uk/nrl

northumbria
UNIVERSITY NEWCASTLE

Integrating Social Circles and Network Representation Learning for Item Recommendation

Yonghong Yu*, Qiang Wang*, Li Zhang†, Can Wang‡, Sifan Wu*, Boyu Qi*, and Xiaotian Wu*

*Tongda College
Nanjing University of Posts and Telecommunications, China
Email: yuyh@njupt.edu.cn
†Department of Computer and Information Sciences
Northumbria University, Newcastle, UK
Email: li.zhang@northumbria.ac.uk
‡School of Information and Communication Technology
Griffith University, Gold Coast, Australia
Email: can.wang@griffith.edu.au

Abstract—With the increasing popularity of social network services, social network platforms provide rich and additional information for recommendation algorithms. More and more researchers utilize trust relationships of users to improve the performance of recommendation algorithms. However, most of existing social-network-based recommendation algorithms ignore the following problems: (1) In different domains, users tend to trust different friends. (2) the performance of recommendation algorithms is limited by the coarse-grained trust relationships. In this paper, we propose a novel recommendation algorithm that integrates social circles and network representation learning for item recommendation. Specifically, we first infer domain-specific social trust circles based on original users' rating information and social network information. Next, we adopt network representation technique to embed domain-specific social trust circle into a low-dimensional space, and then utilize the low-dimensional representations of users to infer the fine-grained trust relationships between users. Finally, we integrate the fine-gained trust relationships into domain-specific matrix factorization model to learn latent user and item feature vectors. Experimental results on real-world datasets show that our proposed approach outperforms traditional social-network-based recommendation algorithms.

I. Introduction

In the era of big data, it becomes more and more difficult to find valuable related information from massive data. Recommender systems [1] infer users' latent preferences by analyzing users' past activities and provide users with personalized recommendation services. Hence, recommender systems have become an effective means to solve the problem of information overload.

As one of the most widely used techniques for recommender systems, collaborative filtering (CF) [2] has achieved great success in E-commerce. However, The problems of data sparsity and cold start significantly hinder the performance of collaborative filtering methods. For example, due to data sparsity, the traditional collaborative filtering algorithms cannot accurately calculate the similarity between users or between items; or latent feature vectors of users and items can not be accurately learned from users' past activities.

The emergence of social networks offers researchers an opportunity to improve the performance of traditional recommendation algorithms by utilizing the rich information of social networks, especially for solving the data sparsity and cold start issues. Social-network-based recommendation algorithms [3]–[8] generally assume that users with trust relations usually share common interests. Typical social-network-based recommendation algorithms include SoRec [3], RSTE [4], SocialMF [6], TrustMF [7] and so on. However, most of existing social-network-based recommendation algorithms ignore that users trust different friends in different domain. Taking an example, we suppose that Jack trusts Tom and may be affected by Tom's recommendations about electronics because Tom is an expert in electronics. But it does not means that Jack also trust Tom on movies and may willing to accept movie-related recommendations from Tom. Moreover, the coarse-grained and sparse social trust relationships significantly degrade the performance of the social-network-based recommendation algorithms. The social-network-based recommendation algorithms generally utilize binary trust value to indicate the degree of trust between two users. If there is a trust connection between two users, then the degree of trust between them is 1, otherwise, the trust degree is 0. Intuitively, the granularity of such a representation is too rough to specify the different trust degrees among users. Further more, many users are very likely to trust one another because of their common neighbors, though they have not built any direct trust connections. In the process of designing recommendation models, we can largely enhance the quality of recommendation algorithms by taking into account such indirect and implicit trust relation. However, such implicit trust relation between users is often neglected in the traditional social-network-based recommendation models.

In addition, in order to model the structures of large scale information networks, a series of graph representation learning algorithms have been proposed [9]–[13]. The graph representation learning algorithms embed a large-scale information net-

work into the low-dimension space. Each node is represented by the low-dimensional feature vectors. Such low-dimensional feature vectors preserve the structures of information networks effectively, which can be applied in various tasks base on graph, such as node classification, clustering, link prediction, and visualization. The classic graph representation learning models include graph factorization [11], DeepWalk [12], LINE [13], and etc. In particular, the LINE model learns the embedded representations of large-scale information networks, by simultaneously preserving the global and local structures for information networks. LINE model is applicable in any kind of large-scale homogeneous information networks, including directed/undirected and weighted/unweighted information networks. Besides, the embedded representations of nodes learned by the LINE model simultaneously preserve the first-order similarity and the second-order similarity between nodes. Intuitively, it can be used to solve the issues of the rough granularity of user trust relation, as well as the only consideration of explicit trust relation modeling in social-network-based recommendation algorithms. However, very limited research work has been proposed to apply the graph representation learning algorithms to improve the recommendation quality of social-network-based recommendation algorithms.

In order to tackle the above problems, we propose a novel recommendation algorithm that integrate social trust circles and network representation learning to boost the traditional social-network-based recommendation algorithms. Specifically, we first infer domain-specific social trust circles based on original users' rating information and social network information. Next, we adopt network representation technique to embed domain-specific social trust circles into low-dimensional spaces. Then, we infer the fine-grained trust relationships using the inferred low-dimensional representations of users, which simultaneously capture the explicit and implicit trust relationships between users. Finally, we integrate the fine-gained trust relationships into domain-specific matrix factorization model to learn user and item latent feature vectors. Experimental results on real-world datasets show that our proposed approach outperforms traditional social-network-based recommendation algorithms.

II. RELATED WORK

In this section, we review the major related work for recommender systems, including traditional collaborative filtering methods, social-network-based recommendation algorithms and network presentation learning techniques.

A. Collaborative Filtering

Collaborative filtering recommendation algorithms (CF) [2] are widely deployed in the modern E-commerce web sites, and have achieve a great success. CF approaches are mainly divided into three categories [1]: memory-based collaborative filtering algorithms, model-based collaborative filtering algorithms and hybrid recommendation algorithms.

The memory-based collaborative filtering recommendation algorithms utilize entire user-item rating matrix for recommen-

dation. The underlying assumption of memory-based methods is that similar users share common interests, and users usually prefer similar items. Typical memory-based collaborative filtering algorithms include user-based collaborative filtering algorithm [14] and item-based collaborative filtering algorithm [15], [16].

The model-based collaborative filtering algorithms adopt statistics and machine learning techniques to learn a prediction model from the user-item rating matrix. The prediction model describes users' behavior patterns, and then is used for item recommendation. Typical model-based filtering approaches include Bayesian networks [2], clustering model [17], [18], latent semantic analysis [19], and restricted Boltzmann machines [20].

Over the last decade, matrix factorization methods [21], [22] have attracted great attention from academia and industry due to their good scalability and accurate prediction ability in dealing with large-scale data. The matrix factorization methods assume that only a few latent factors contribute to the preferences of users and the characteristics of items. Therefore, the matrix factorization methods map users and items to a low-dimensional latent factor space, such that the correlations between users and items can be directly calculated using latent user and item feature vectors. Typical matrix factorization methods include NMF [23], PMF [22], SVD++ [24], and MMMF [25] and so on.

In fact, the data sparsity of user-item rating information seriously limits the performance of collaborative filtering algorithms. For example, the item-based collaborative filtering algorithm can not accurately find similar items for the target item that has few ratings or even no ratings. Also, it is hard for the matrix factorization methods to accurately learn the latent feature vectors for new registered users or new added items.

B. Social-network-based Recommendation Algorithm

The emergence of social networks provides an opportunity to alleviate the problem of data sparse and cold start in collaborative filtering algorithms. By utilizing the rich information of social networks, i.e., trust relationships, user comments and item descriptions, researchers have proposed several typical social-network-based recommendation algorithms, such as SoRec [3], RSTE [4], SocialMF [6], TrustMF [7] and so on.

Ma et al. [3] proposed a recommendation algorithm based on probabilistic matrix factorization model called SoRec. SoRec integrates rating information with social trust relationships, and two different types of information sources are fused by sharing the user latent feature matrix between the ratings and the social trust relationships. To more accurately model the process of decision making, Ma et al. [4] proposed RSTE. RSTE assumes that the final decision is a trade-off between the user's own preferences and his friends' preferences. In [6], Jamali et al. proposed SocialMF, which integrates trust propagation mechanism into the matrix factorization model. The trust propagation mechanism is particularly effective to deal with the cold start problem because the latent feature vector of a new registered user may be inferred from the feature vectors

of the user's most similar neighbors, whose feature vectors can be accurately learned from their rating information. In [7], Yang et al. integrated rating information with trust information, and proposed a recommendation algorithm called TrustMF. TrustMF assumes that users are influenced by the rating and comment information of their trusted friends. Meanwhile, TrsutMF also assumes that users' own ratings and comments can also affect other users' decisions. The experimental results of the mentioned above social-network-based recommendation algorithms report that social network information is helpful to improve the performance of traditional collaborative filtering recommendation algorithms, especially to alleviate the cold start problem. However, all the above social-network-based recommendation methods only utilize the coarse-grained and sparsity trust relationships, i.e., the observed trust relationships, to boost the recommendation performance. In [26], Yang et al. proposed a recommendation algorithm based on the social network circles, called CircleCon, which is most related to our work. CircleCon first deduces domain-specific social trust relation networks, then SocialMF algorithm is employed to learn the user latent feature vectors and the item latent feature vectors in each domain. It should be noted that only observed and binary trust relationships are utilized in the SocialMF component. Differ from CircleCon, our proposed method not only considers the social network circle in different domains, but also infers the domain-specific fine-grained and dense trust relationships using the low-dimensional representations of users, which are learned by utilizing the network representation technique. The inferred fine-grained trust relationships simultaneously capture the global and local structures of social network circles, which indicate the implicit and explicit trust relationships, respectively.

C. Network Representation Learning

Network representation learning [9], [10], also known as network embedding or graph embedding, aims to embed large-scale information networks into low-dimensional spaces, such that each node of information network is represented as a low-dimensional vector. More importantly, the low-dimensional vectors encode the structures of large-scale information networks. The learned low-dimensional vectors can be utilized in many machine learning tasks, such as visualization, node classification, link prediction and community discovery. Typical network representation learning methods include Graph Factorization [11], DeepWalk [12], LINE [13], etc. Graph factorization [11] uses matrix factorization to learn the embedded representations of large-scale networks. However, since the objective function of the matrix factorization employed in graph factorization is not designed for the network, the global structure of information network can not be captured. Meanwhile, graph factorization model is only suitable for undirected information networks. DeepWalk [12] adopts random walk algorithm to learn the embedded representations. But, DeepWalk does not clearly describe what network properties are preserved. DeepWalk model is only applicable to unweighted information networks. In LINE [13], the objective function

of LINE preserves local and global structures of large-scale information networks. In addition, LINE model employs edge-based sampling strategy to deal with the limitations of the classical stochastic gradient descent algorithm (SGD). LINE model is suitable for large-scale homogeneous information networks, including directed/undirected and weighted/unweighted information networks.

Based on the advantages of LINE model, we adopt LINE model to learn the embedded representations of user vertices in the social network, and integrate the fine-grained trust values inferred from the embedded representations of users into the social-network-based recommendation algorithm to improve the recommendation performance.

III. PRELIMINARIES

A. Problem Description

Social-network-based recommender systems often contain two different types of data sources: user-item rating matrix and social network information. User-item rating matrix $R \in \mathbb{R}^{N \times M}$ consists of two sets of entities: a set of N users $U = \{u_1, u_2, ..., u_N\}$ and a set of M items $I = \{i_l, i_2, ..., i_M\}$. Each entry r_{ui} of R represents the rating of user u on item i. In principle, the rating $r_u i$ can be any real number, but the rating typically is an integer, and $r_{ui} \in \{0, 1, 2, 3, 4.5\}$, where 0 indicates that the user has not rated the item. A higher rating means that the user is more satisfied with the current item. Since users usually rate only a small fraction of items, user-item rating matrix R is extremely sparse. For example, there are 93% and 95% missing ratings in MovieLens100K and MovieLens1M datasets, respectively. The sparsity of user-item rating matrix leads to poor recommendation quality.

Social network information is represented as a directed social relationship graph $G = (U, E)$, where U is the user set and the edge set E represents the social trust relationships between users. $t_{u,v} \in [0.1]$ indicates the trust degree between user u and v, and $t_{u,v} = 0$ means that no trust relationship is established between user u and v. All trust relationships constitute the trust matrix T. It should be noted that the trust matrix T is usually asymmetric because the trust relationships between users are often not mutual.

The goal of social-network-based recommendation system is to provide users with ranked lists of items by utilizing both rating and social network information.

B. Matrix Factorization

Matrix factorization (MF) is one of well-known recommendation methods and widely deployed in E-commerce. Matrix factorization maps users and items to a low-dimensional latent factor space, such that the correlations between users and items can be directly calculated using latent user and item feature vectors. Formally, given the user latent feature matrix $P \in \mathbb{R}^{K \times N}$ and the item latent feature matrix $Q \in \mathbb{R}^{K \times M}$ respectively ($K \ll \min\{M, N\}$), where K is the dimension of the latent feature vectors, MF learns the latent feature matrix P and Q by minimizing the sum-of-squared-error objective function:

$$\min_{P,Q} \frac{1}{2} \sum_{(u,i)\in\Omega} (r_{ui} - p_u^T q_i)^2 + \frac{\lambda_1}{2}||P||_F^2 + \frac{\lambda_2}{2}||Q||_F^2, \quad (1)$$

where Ω is the set of observed (user, item) pairs in R, p_u and q_i represent the latent feature vectors of user u and item i, respectively. And $||.||_F^2$ is the Frobenius norm. Regularization terms $||P||_F^2$ and $||Q||_F^2$ are used to avoid overfitting. λ_1 and λ_2 are regularization parameters used to control the influence of the regularization terms.

IV. INTEGRATING SOCIAL CIRCLES AND NETWORK REPRESENTATION LEARNING FOR ITEM RECOMMENDATION

Social-network-based recommendation algorithms generally integrate the original trust relationships of social networks into the classical matrix factorization models. They assume that users with trust relationships have common interests and preferences. However, there are several issues in the process of integrating original trust relationships into recommendation models: 1) in different domain, users often trust different friends. 2) traditional social-network-based recommendation algorithms use the coarse-grained trust values, i.e. binary trust values, to represent the degrees of trust between users. The granularity of trust value is too rough to distinguish the different degree of trust among users. 3) only observed trust relationships are considered, and implicit trust relationships are often neglected in traditional social-network-based recommender systems. The observed trust relationships only capture the local structure of social network, but implicit trust relationships encode global structure of social network. Many users are highly probable to have large trust degrees between one another because of their common neighbors, though they have not made any direct trust link.

In this paper, the observed trust relation is called the first-order trust (i.e., explicit trust relationships), and the trust relation induced by the neighborhood structure is named the second-order trust (i.e., implicit trust relationships). The consideration of the second-order trust relationships will greatly improve the quality of recommendation algorithms in the process of recommendation modeling.

In the following sections, we first present the process of learning users' embedded representations by utilizing LINE model, and then describes the recommendation algorithm framework of integrating social circles and network representation learning for item recommendation. Finally, we introduce the recommendation model and parameter learning process.

A. Learning Embedded Representations of Users

LINE [13] is an important representative of network representation learning technique. In our proposed recommendation model, we generally apply LINE model to learn users' embedded representations for each social circle, whose inference process is described in Section IV-B.

In each social circle, we define the joint probability distribution between user trust pair (u, v) to model the first-order trust between them. Formally,

$$p_1(x_u, x_v) = \frac{1}{1 + exp(-y_u^T y_v)}, \quad (2)$$

where $y_u, y_v \in R^{d_1}$ is the low-dimensional vector representation of vertex x_u. The empirical distribution between vertices x_u and x_v is defined as follows:

$$\hat{p_1}(x_u, x_v) = \frac{w_{uv}}{W}, \quad (3)$$

where $W = \sum_{(u,v)\in E} w_{uv}$, and w_{uv} is the weight of the edge (u, v). To preserve the first-order trust in social circle, we minimize the KL-divergence between the joint probability distribution and the empirical probability distribution. Formally, the objective function is defined as:

$$O_1 = - \sum_{(u,v)\in E} w_{uv} log p_1(x_u, x_v) \quad (4)$$

The concept of implicit trust assumes that two users who share similar neighbors are highly probable to have large trust degrees between them. Specifically, each user vertex is also treated as a specific "context" and users with similar "contexts" trust each other. Therefore, each user vertex plays two roles: the user vertex itself and the specific "context" of other user vertices. For each directed user edge (u, v), the probability distribution of generating "context" x_v from user vertex x_u is defined as:

$$P_2(x_v|x_u) = \frac{exp(y_v^{*T} y_u)}{\sum_{k=1}^{|U|} exp(y_k^{*T} y_u)}, \quad (5)$$

where $|U|$ is the number of user vertices or "contexts", and $y_v^* \in R^{d_2}$ is the low-dimensional representation of x_v as "context". The empirical distribution of "context" x_v generated by user vertex x_u is:

$$\hat{P}_2(x_v|x_u) = \frac{w_{uv}}{d_u}, \quad (6)$$

where d_u is the out-degree of user vertex x_u, i.e. $d_u = \sum_{v\in N(u)} w_{uv}$, where $N(u)$ is the set of neighbors of x_u.

Similarly, in order to preserve the second-order trust, the following objective function is obtained by utilizing KL-divergence:

$$O_2 = - \sum_{(u,v)\in E} w_{uv} log p_2(x_v|x_u) \quad (7)$$

LINE model minimizes the objective functions O_1 and O_2 separately, and learns two low-dimensional representations for each user vertex, which encode the first-order and second-order trust, respectively. Then, it combines the two low-dimensional representations as one low-dimensional feature vector to simultaneously preserve the local and global structure of social circle. Therefore, each vertex x_u can be represented as $y_u \in R^d$, where $d = d_1 + d_2$.
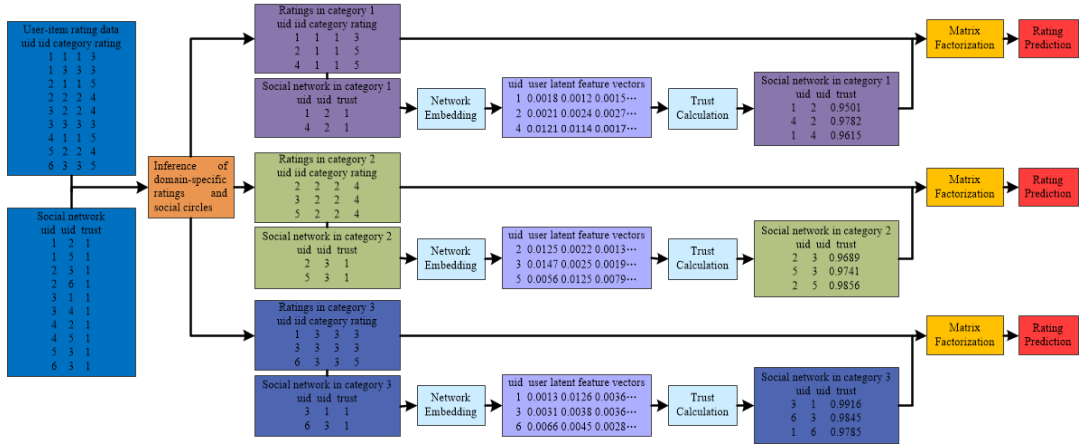
Fig. 1. The Framework of Our Proposed Recommendation Algorithm

B. The Framework of Integrating Social Circles and Network Representation Learning for Item Recommendation

The framework of integrating social circles and network representation learning for item recommendation is showed in Fig. 1. It is mainly composed of the inference of domain-specific ratings and social circles, learning embedded representations of users, computing the fine-grained trust values, matrix factorization with the fine-grained trust values, and rating prediction components.

- Inference of domain-specific ratings and social circles: we infer the domain-specific ratings by splitting the original rating matrix R according to the categories of items. For example, in the Epinions dataset, the ratings of users on items are expressed in the form of 4 tuples $(uid, iid, category, rating)$, where $category$ represents the category of the rated item. The original rating matrix R can be divided into $R^{c_1}, R^{c_2}, ... R^{c_l}$ according to categories, where l is the number of item categories in dataset. After inferring the domain-specific ratings, the social circle for each domain is deduced based on the principle of co-occurrence of ratings and trust relationships. Formally, in domain c, the trust value t_{uv}^c between users u and v is defined as follow:

$$t_{uv}^c = \begin{cases} t_{uv}, & t_{uv} \neq 0 \wedge N_u^c > 0 \wedge N_v^c > 0 \\ 0, & otherwise, \end{cases} \quad (8)$$

where N_u^c and N_v^c represent the number of ratings of users u and v in R^c, respectively. t_{uv} is the trust value between users u and v in the original social network. For each domain, all social trust relationships deduced based on the above principle constitute the domain-specific social circle, denoted as social trust matrix T^c.

- Learning embedded representations of users: in this component, for each social circle, we learn user embedded presentations by utilizing network presentation learning technique, which are described in Section IV-A.

- Computing the fine-grained trust values: we utilize the inner product of users' embedded presentations to compute the fine-grained trust among users. Formally,

$$s_{uv}^c = (y_u^c)^T y_v^c \quad (9)$$

Compared with the coarse-grained trust value t_{uv}^c, it should be noted that the fine-grained trust value s_{uv}^c is more informative, and can accurately distinguish the different degree of trust among users. Moreover, the fine-grained trust measure encodes both the first-order and second-order trust relationships among users since users' embedded presentations capture the local and global structure of social circle.

- Matrix factorization with the fine-grained trust values: for each domain, this component integrates the fine-grained trust relationships into classical matrix factorization model to learn domain-specific latent user feature matrix P^c and latent item feature matrix Q^c. The process of learning P^c and Q^c is presented in Section IV-C.

- Rating prediction: we use the dot product between latent user feature vector p_u^c and item feature vector q_i^c to predict the missing rating $\hat{r}_{ui}^c$:

$$\hat{r}_{ui}^c = (p_u^c)^T q_i^c \quad (10)$$

C. Model and Parameters Learning

Generally, for each domain, we integrate the fine-grained trust relationships into classical matrix factorization model to learn latent user and item feature vectors, and use the learned feature vectors to predict the missing ratings. In addition, we incorporate trust propagation mechanism into matrix factorization, which is similar to SocialMF [6].

Without loss of generality, we map the ratings r_{ui}^c to the interval [0,1] using the function $f(x) = (x - minRating)/(maxRating - minRating)$, where $maxRating$ and $minRating$ are the maximum and minimum ratings in recommender systems, respectively. Meanwhile, we use logistic function $g(x) = 1/(1 + e^{-x})$ to

limit the predicted ratings $\widehat{r}_{ui}^c$ within the range [0,1]. The objective function for domain c is defined as:

$$L^c = \frac{1}{2} \sum_{(u,i)\in\Omega} (r_{ui}^c - g((p_u^c)^T q_i^c))^2 + \frac{\lambda_1}{2}||P^c||_F^2$$
$$+ \frac{\lambda_2}{2}||Q^c||_F^2 + \frac{\lambda_3}{2}\sum_{u=1}^{N^c}||p_u^c - \sum_{v\in F^c(u)} s_{uv}^c p_v^c||_F^2, \quad (11)$$

where Ω^c is observed (user, item) pairs in R^c. $F^c(u) = \{v|s_{uv}^c > \delta\}$ is the set of users whom user u trusts in the domain c.

Unlike SocialMF, our proposed recommendation model not only considers the differences of trust degree among users in different domains, but also considers the fine-grained trust values when integrating trust propagation mechanism.

Similar to matrix factorization algorithms, the stochastic gradient descent algorithm (SGD) is applied to seek a local minimum of the objective function L^c.

V. EMPIRICAL ANALYSIS

In this section, we conduct several experiments on real datasets to compare the performance of our proposed recommendation algorithms with other state-of-the-art methods.

A. Dataset

We choose Epinions dataset to evaluate the performance of our proposed methods. Epinions dataset contains user ratings, social relationships, item categories etc. Epinions dataset used in our experiments is provided by the authors of reference [27]. It contains 922267 ratings, 22166 users, 296277 items, and 355813 trust relationships. The sparse level of the user-item rating matrix is 99.986%. The items in Epinions are divided into 27 categories, and the distribution of the number of categories involved by users is plotted in Fig. 2.
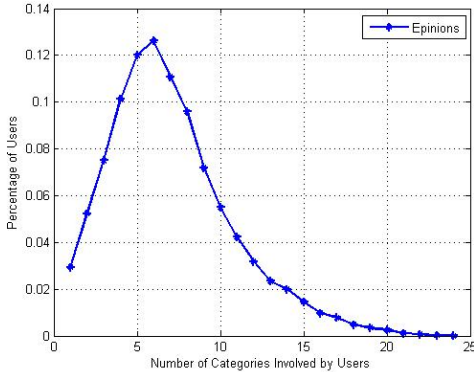

Fig. 2. Distribution of the number of categories involved by users

As shown in Fig. 2, the average number of item categories involved by users is relatively small. Only 21.7% users involve in more than 10 categories, which means that most users only interested in a few categories. This implies that a user who is trusted by other users is not always affects others decision making. It also to some extent confirms our assumption that a user generally trust different users in different domains. We evaluate our proposed recommendation algorithms on "Books", "Games" and "Sports & Outdoors" sub-datasets, which represent large, medium and small datasets, respectively.

B. Evaluation Metric

We use $RMSE$, which is widely used evaluation metric in recommender systems, to evaluate the performance of recommendation algorithms. $RMSE$ is defined as:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} |r_{ui} - \hat{r}_{ui}|^2}{|R_{test}|}}, \quad (12)$$

where r_{ui} and $\hat{r}_{ui}$ represent the actual rating and the predicted rating, respectively. $|R_{test}|$ represents the number of records in the test dataset. the lower the $RMSE$, the better the recommendation algorithm.

C. Experimental Settings

In order to evaluate the effectiveness of our proposed recommendation algorithm, we select following recommendation algorithms as comparison methods:

- PMF: PMF [22] was proposed by Mnih and Salakhutdinov. PMF can be regarded as the probability extension of SVD model.
- SoRec: SoRec [3] simultaneously factorizes the user rating matrix and user trust matrix, and fuses the rating information and social network information by sharing the user latent feature matrix.
- RSTE: RSTE [4] assumes that the final decision is a trade-off between the user's own preferences and his friends' preferences.
- SocialMF: SocialMF [6] integrates trust propagation mechanism into PMF to improve the accuracy of the recommendation algorithm.
- TrustMF: TrustMF [7] performs matrix factorization on user trust matrix to map users into two different latent feature spaces: the truster feature space and the trustee feature space.

We randomly extract 80% of the user-item rating data as the training dataset, and the remaining 20% as the test dataset. This random extraction is performed 5 times independently, and the average results on 5 test datasets are reported. In order to make a fair comparison, we set the parameters of each algorithm according to respective references or based on our experiments. Under these parameter settings, each comparison algorithm achieves the optimal performance. In PMF, $\lambda_U = \lambda_V = 0.001$; in SoRec, $\lambda_U = \lambda_V = \lambda_Z = 0.001$, $\lambda_C = 1$; in RSTE, $\lambda_U = \lambda_V = 0.001$, $\alpha = 0.4$; in SocialMF, $\lambda_U = \lambda_V = 0.001$, $\lambda_T = 1$; in TrustMF, $\lambda = 0.001$, $\lambda_T = 1$; For our proposed method, $\lambda_1 = \lambda_2 = 0.001$, $\lambda_3 = 1$. It should be noted that for the classical social-network-based recommendation models, we utilize all the social relationships contained in Epinions to train recommendation model. While for our proposed method, we employ inferred domain-specific social circle to train recommendation model.

D. Performance Comparison

For our proposed algorithm, we set the dimension of embedded presentation $d = 128$ and $\delta = 0.8$. The experimental results of all comparison algorithms on three datasets are shown in Table I, Table II and Table III.

TABLE I
PERFORMANCE COMPARISONS ON BOOKS

Recommendation Algorithm	RMSE(K = 10)	RMSE(K = 20)
PMF	2.997457	3.095979
SoRec	1.002975	1.088180
RSTE	1.022563	1.084097
SocialMF	0.967169	1.071207
TrustMF	1.113678	1.264988
Our Method	0.959160	1.061402

TABLE II
PERFORMANCE COMPARISONS ON GAMES

Recommendation Algorithm	RMSE(K = 10)	RMSE(K = 20)
PMF	2.007327	2.115031
SoRec	1.099372	1.230581
RSTE	1.163706	1.175594
SocialMF	1.074102	1.207000
TrustMF	1.324873	1.498723
Our Method	1.054989	1.156762

TABLE III
PERFORMANCE COMPARISONS ON SPORTS & OUTDOORS

Recommendation Algorithm	RMSE(K = 10)	RMSE(K = 20)
PMF	3.093023	3.153216
SoRec	1.071452	1.197364
RSTE	1.088458	1.127339
SocialMF	1.041038	1.168522
TrustMF	1.156120	1.306297
Our Method	1.027257	1.117535

From Table I, Table II and Table III, we have the following observations: (1) All social-network-based recommendation algorithms outperform PMF, which only utilizes ratings to learn latent feature vectors. This observation indicates that social network information indeed is beneficial to recommendation algorithms. (2) On three datasets, our proposed algorithm is consistently superior to other comparisons, which demonstrates the effectiveness of our proposed method. when $K = 10$, compared with the optimal results among PMF, SoRec, RSTE, SocialMF and TrustMF, the improvements of our proposed algorithm on "Books", "Games" and "Sports & Outdoors" are 0.83%, 1.78% and 1.32%, respectively. This observation confirms our assumption that integrating the fine-grained trust relationships, which encode both first-order and second-order trust relationships, can improve the performance of social-network-based recommendation algorithms. (3) In addition, the performance of all comparison algorithms degrades with the increasing of K, which implys that increasing the dimension of latent feature vectors can not effectively improve the performance of matrix factorization

based recommendation models. This is because that only a small number of latent factors contribute users' preferences and items' characteristics, which is underlying assumption of matrix factorization models.

E. Impact of Parameter δ

In our proposed algorithm, the trust threshold δ is an important parameter that affects the performance of our proposed recommendation algorithm. Large δ means that filters out the weak user trust relationships, and integrates strong trust relationships. On the contrary, small δ means that integrates relatively weak trust into our proposed recommendation model. In this section, we perform a set of experiments to investigate the impact of parameter δ on recommendation performance. We set δ be 0.8, 0.85, 0.9 and 0.95, and observe the change trends of $RMSE$. Meanwhile, we set $d = 128$ and $K = 10$. The experimental results are shown in Fig. 3.
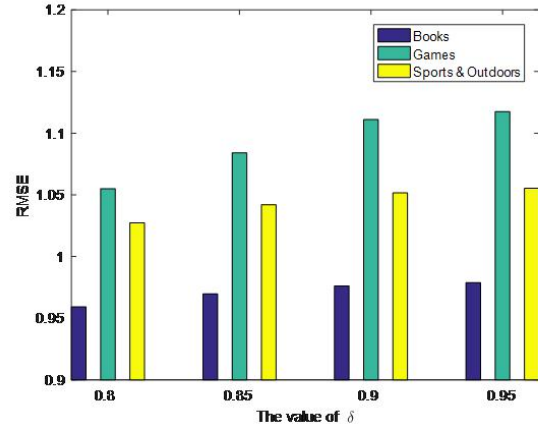

Fig. 3. Impact of parameter δ

As shown in Fig. 3, the parameter δ really affects the performance of our proposed algorithm. On the three datasets, the values of $RMSE$ show a similar trend: as δ increases, $RMSE$ increases gradually, and the accuracy of recommendation algorithm decreases. This observation indicates that the relatively small δ is beneficial to our proposed recommendation model. This reason is that: increasing the value of δ filters out most user trust relationships, including weak and relatively strong trust relationships, which results in extremely sparse user trust relationships and hurts the performance of the recommendation algorithm. Hence, integrating the fine-grained, relatively strong and dense trust relationships is more conducive to improving the performance of our proposed model.

F. Impact of Parameter d

In this section, we vary the value of d from 128 to 512, and investigate the impact of parameter d on recommendation quality. We set $\delta = 0.8$ and $K = 10$. The experimental results are plotted in Fig. 5.

As we can see, our proposed recommendation method achieves its best performance when d is equal to 128. This
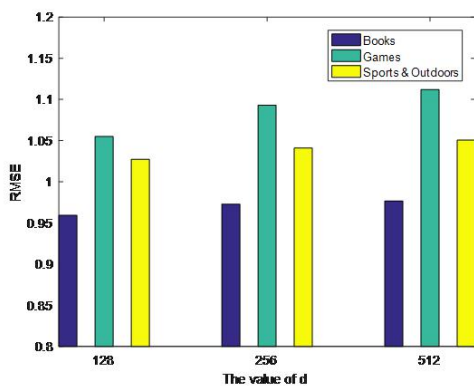
Fig. 4. Impact of parameter d

observation indicates that increasing the value of d can not improve the performance of our proposed method. A possible reason is that: although large dimensional features learned from network representation learning model enhance the representation capacity of network representation learning model, it may also introduce noises, which affect the inference of social trust in recommendation model.

VI. CONCLUSION

To solve the problems of the coarse-grained trust relationships as well as users usually trust different friends in different domains, we propose a novel recommendation algorithm that integrates social circles and network representation learning for item recommendation. Specifically, we first infer domain-specific social trust circles based on original users' rating information and social network information. Next, we adopt network representation technique to embed domain-specific social trust circle into a low-dimensional space, and then utilize the low-dimensional representations of users to infer the fine-grained trust relationships between users. Finally, we integrate the fine-gained trust relationships into domain-specific matrix factorization model to learn latent user and item feature vectors. Experimental results on real-world datasets show that our proposed approach outperforms traditional social-network-based recommendation algorithms.

ACKNOWLEDGMENTS

REFERENCES

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[2] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *UAI*, 1998.

[3] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM*. ACM, 2008, pp. 931–940.

[4] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *SIGIR*, 2009, pp. 203–210.

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

[6] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *RecSys*, 2010.

[7] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social collaborative filtering by trust," in *IJCAI*, 2013, pp. 2747–2753.

[8] Y. Yu, Y. Gao, H. Wang, and R. Wang, "Joint user knowledge and matrix factorization for recommender systems," *World Wide Web*, vol. 21, no. 4, pp. 1141–1163, 2018.

[9] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[10] H. Cai, V. W. Zheng, and K. Chang, "A comprehensive survey of graph embedding: problems, techniques and applications," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[11] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *WWW*. ACM, 2013, pp. 37–48.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*. ACM, 2014, pp. 701–710.

[13] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWWW*. International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *CSCW*, 1994.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *WWW*. ACM, 2001, pp. 285–295.

[16] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[17] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *SIGIR*, 2005.

[18] Y. Yu, C. Wang, Y. Gao, L. Cao, and X. Chen, "A coupled clustering approach for items recommendation," in *PAKDD*, 2013, pp. 365–376.

[19] T. Hofmann, "Latent semantic models for collaborative filtering," *TOIS*, vol. 22, no. 1, pp. 89–115, 2004.

[20] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *ICML*, 2007.

[21] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[22] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *NIPS*, 2007, pp. 1257–1264.

[23] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001.

[24] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *KDD*, 2008, pp. 426–434.

[25] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *NIPS*, 2004.

[26] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *KDD*, 2012, pp. 1267–1275.

[27] J. Tang, H. Gao, and H. Liu, "mTrust: Discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 93–102.