IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 4, 2019, accepted April 30, 2019, date of publication May 10, 2019, date of current version May 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2916186

Network Representation Learning Enhanced Recommendation Algorithm

QIANG WANG[1], YONGHONG YU[1], HAIYAN GAO[1], LI ZHANG[2], YANG CAO[1], LIN MAO[1], KAIQI DOU[1], AND WENYE NI[1]

[1]Tongda College, Nanjing University of Posts and Telecommunications, Yangzhou 225127, China
[2]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

Corresponding author: Yonghong Yu (yuyh@njupt.edu.cn)

This work was supported in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 17KJB520028, in part by the NUPTSF under Grant NY217114, in part by the Tongda College, Nanjing University of Posts and Telecommunications under Grant XK203XZ18002, and in part by the Qing Lan Project of Jiangsu Province.

ABSTRACT With the popularity of social network applications, more and more recommender systems utilize trust relationships to improve the performance of traditional recommendation algorithms. Social-network-based recommendation algorithms generally assume that users with trust relations usually share common interests. However, the performance of most of the existing social-network-based recommendation algorithms is limited by the coarse-grained and sparse trust relationships. In this paper, we propose a network representation learning enhanced recommendation algorithm. Specifically, we first adopt a network representation technique to embed social network into a low-dimensional space, and then utilize the low-dimensional representations of users to infer fine-grained and dense trust relationships between users. Finally, we integrate the fine-grained and dense trust relationships into the matrix factorization model to learn user and item latent feature vectors. The experimental results on real-world datasets show that our proposed approach outperforms traditional social-network-based recommendation algorithms.

INDEX TERMS Network representation learning, recommendation algorithm, matrix factorization, social network.

I. INTRODUCTION

In the era of big data, it becomes increasingly difficult to find valuable related information from massive unstructured data. Recommender systems [1] infer users' latent preferences by analyzing their past activities and provide them with personalized recommendation services. Therefore recommender systems have become an effective means to solve the problem of information overload. In recent years, such research directions have drawn great attention from academia and industry. Typical applications of recommender systems include Amazon's product recommendation, Netflix's movie recommendation, last.fm's music recommendation, LinkedIn's friend recommendation, and Google News's news recommendation.

Collaborative filtering (CF) [2] is the most widely used recommendation technique in the research of recommender systems. However, the problems of data sparsity and cold start have significantly negative impact on the performance of collaborative filtering methods. As an example, owing to data sparsity, traditional collaborative filtering algorithms cannot accurately calculate the similarities between users or between items; or cannot accurately learn latent user and item feature vectors from users' past activities.

The emergence of social networks brings an opportunity to alleviate the problems of data sparsity and cold start in traditional collaborative filtering algorithms. Some researchers utilize the rich information contained in social networks to propose some social-network-based recommendation algorithms. Typical social-network-based recommendation algorithms include SoRec [3], RSTE [4], SocialMF [5], TrustMF [6] and so on. Social-network-based recommendation algorithms generally assume that users with trust relations usually share common interests. However, in the original social network, the trust relationship is usually binary, that is, only 0 or 1 is used to denote the trust relationship between users where 1 denotes there is a trust relation between two users, and the degree of trust is 1 and 0 indicates that there is no trust relationship between users. Intuitively, the granularity of such a representation is too coarse to specify the different levels of trust among users.

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Fiore.

In fact, many users are very likely to trust one another because of their shared connections, although they have not built any direct trust connections. In the process of designing recommendation models, the quality of recommendation algorithms can be enhanced by considering such indirect and implicit trust relationships. However, such implicit trust relationships between users are often ignored in the traditional social-network-based recommendation models.

In order to tackle the above problems, this research proposes a network representation learning enhanced recommendation algorithm. Specifically, we first adopt a network representation technique [7] to embed social network into a low-dimensional space, and then utilize the low-dimensional representations of users to infer fine-grained and dense trust relationships between users. Finally, we integrate the fine-grained and dense trust relationships into the social-network-based recommendation model to learn latent feature vectors of users and items more precisely. The empirical results on real-world datasets indicate that our proposed approach outperforms traditional social-network-based recommendation algorithms.

II. RELATED WORK

In this section, we review the state-of-the-art related work for recommender systems, including social-network-based recommendation algorithms and network representation learning techniques.

A. SOCIAL-NETWORK-BASED RECOMMENDATION ALGORITHMS

Although collaborative filtering algorithms [8]–[13], including matrix factorization based methods [14]–[18], have achieved great success in E-commerce, the problems of data sparsity and cold start significantly hinder the performance of collaborative filtering methods. The emergence of social networks provides an opportunity for collaborative filtering to alleviate the problem of data sparse and cold start. By utilizing the rich information of social networks, i.e., trust relationships, user comments and item descriptions, researchers have proposed several social-network-based recommendation algorithms [3]–[6], [19]. Typical social-network-based recommendation algorithms include SoRec [3], RSTE [4], SocialMF [5], TrustMF [6] and so on.

In [3], Ma *et al.* proposed a social-network-based recommendation algorithm, namely SoRec, in which rating information and social trust information are fused by sharing the user latent feature matrix. In order to more accurately reflect the real-world recommendation process, Ma *et al.* [4] further proposed RSTE, which combines users' own preferences and their trusted friends' preferences using a weighted parameter. In [5], Jamali and Ester proposed SocialMF, which integrates the trust propagation mechanism into the matrix factorization model to boost the recommendation quality. For alleviating the cold start problem, SocialMF is particularly effective since the latent feature vectors of cold start users may be inferred from the latent feature vectors of their most trusted neighbors who have enough rating information for the matrix factorization model to learn their latent feature vectors. In addition, Yang *et al.* [6] proposed a social-network-based recommendation algorithm, called TrustMF, which combines sparse ratings and social trust relationships to improve the recommendation quality. TrustMF assumes that users are influenced by the rating and comment information of their trusted friends, and users' own ratings and comments also affect other users' decisions. Recently, Yu *et al.* [19] proposed a novel social-network-based recommendation algorithm, named UKMF, which integrates social network information, rating information and users' own knowledge. UKMF assumes that the degree of social influence is different for users with different levels of knowledge, and that users' own knowledge affects the process of their rating-making.

However, the above social-network-based recommendation models generally utilize binary trust values to indicate the degree of trust between two users. The granularity of such a representation is too coarse to specify the different levels of trust among users. Moreover, typical social-network-based recommendation models only integrate observed explicit trust relationships, and ignore the implicit trust relationships. Unlike the aforementioned methods, in this study, we utilize a network representation learning technique to learn the low-dimensional representations of users from the social network, then use the low-dimensional representations to infer the fine-grained trust relationships among users, which simultaneously encode explicit and implicit trust relationships. The fine-grained trust relationships are then integrated into the classic matrix factorization model to boost the recommendation quality.

B. NETWORK REPRESENTATION LEARNING TECHNIQUES

Network representation learning techniques [20], [21] embed the large-scale information network into the low-dimensional space, and each network node is represented as a low-dimensional vector. The low-dimensional representations of nodes learnt from network representation learning techniques can effectively preserve the local and global structures of the large-scale information network. Therefore, network representation learning techniques play an important role in machine learning tasks, such as node classification [22], visualization [23] and link prediction [24]. Typical network representation learning methods include Graph Factorization [25], DeepWalk [26], LINE [7], node2vec [27], etc.

Graph factorization [25] uses matrix factorization to learn the embedded representations of large-scale information networks. However, since the objective function of the matrix factorization employed in graph factorization is not designed for the information networks, the global structures of information networks cannot be captured. Meanwhile, such a graph factorization model is only suitable for undirected information networks. DeepWalk [26] adopts a random walk algorithm to learn the embedded representations. But, DeepWalk does not clearly describe what network properties are preserved. The DeepWalk model is only applicable to unweighted information networks. In [7], Tang *et al.* proposed the LINE model, which learns the embedded representations of users, and preserves the local and global structures of large-scale information networks in the corresponding embedded representations. In addition, the LINE model employs the edge-based sampling strategy to deal with the limitations of the classical stochastic gradient descent algorithm (SGD). The LINE model is suitable for large-scale homogeneous information networks, including directed/undirected and weighted/unweighted information networks. In [27], Grover and Leskovec proposed node2vec, an algorithmic framework for learning continuous feature representations for nodes in networks. The node2vec model learns a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes. Particularly, they defined a flexible notion of a node's network neighborhood and designed a biased random walk procedure to explore diverse neighborhoods. In addition, Qiu *et al.* [28] provided a theoretical analysis of four impactful network embedding methods, i.e. DeepWalk [26], LINE [7], PTE [29] and node2vec [27], and shown that the aforementioned four models with negative sampling can be unified into the matrix factorization framework with closed forms. Moreover, they proposed the NetMF method as well as its approximation algorithm for computing network embedding.

Besides the above mentioned network representation learning models that focus on homogeneous networks, some researchers also proposed several network representation learning models for heterogeneous networks. For example, Dong *et al.* [30] formalized the heterogeneous network representation learning problem, and developed effective and efficient network embedding frameworks, i.e. metapath2vec and metapath2vec++, for preserving both structural and semantic correlations of heterogeneous networks. In [31], Chen *et al.* proposed a novel heterogeneous information network embedding model called PME. The PME model learns a distance metric to preserve both the first-order and the second-order proximities in a unified way, and introduces distinct latent spaces to model objects and relations to alleviate the potential geometrical inflexibility of existing metric learning approaches.

Network representation learning techniques have shown great potential in the community of recommender systems. For instance, Xie *et al.* [32] proposed a graph-based embedding model, called GE, to jointly capture the sequential effect, geographical influence, temporal cyclic effect and semantic effect in a unified way by embedding the four corresponding relational graphs (POI-POI, POI-Region, POI-Time and POI-Word) into a shared low dimensional space. The underlying network representation learning model adopted by GE is the LINE model. Since the embedded representations learnt by the LINE model preserve the local and global structures of large-scale information networks, we aim to improve the recommendation performance of traditional recommendation algorithms by simultaneously considering explicit and implicit trust relationships among users, which are captured by the local and global structures, respectively. Hence, we adopt the LINE model to learn the embedded representations of users from social networks, 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. The 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_{ui} 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, the 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 the 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 users u and v, and $t_{u,v} = 0$ means that no trust relationship is established between users 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 systems is to provide users with ranked lists of items by utilizing both rating and social network information.

B. MATRIX FACTORIZATION

Matrix factorization (MF) [14] 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 $U \in \mathbb{R}^{K \times N}$ and the item latent feature matrix $V \in \mathbb{R}^{K \times M}$ respectively ($K \ll \min\{M, N\}$), where K is the dimension of the latent feature vectors, MF learns the latent feature matrices U and V by minimizing the following sum-of-squared-error objective function:

$$\min_{U,V} \frac{1}{2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{ui}^{R}(r_{ui} - u_u^T v_i)^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_V}{2}||V||_F^2, \tag{1}$$

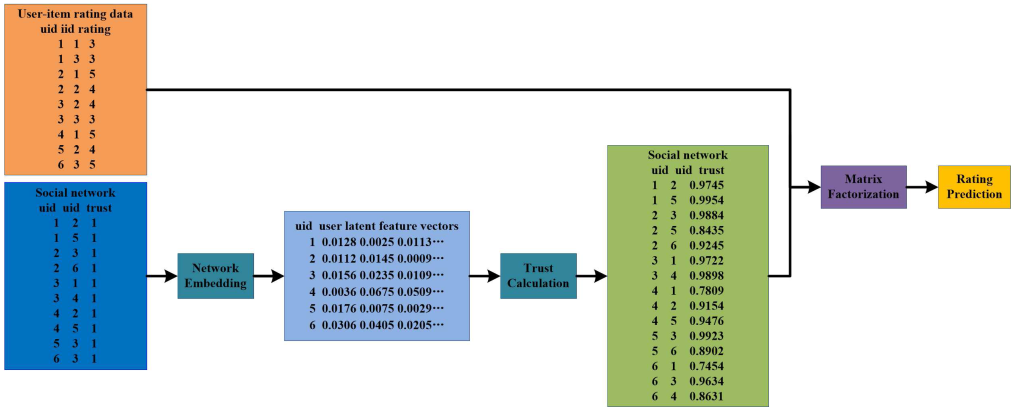


FIGURE 1. The framework of network representation learning enhanced recommendation algorithm.

where I_{ui}^R is the indicator function. For instance, when user u rates item i, it is assigned 1; otherwise, it is 0. u_u and v_i represent the latent feature vectors of user u and item i, respectively. And $||.||_F^2$ is the Frobenius norm. Regularization terms $||U||_F^2$ and $||V||_F^2$ are used to avoid overfitting. λ_U and λ_V are regularization parameters used to control the influence of the regularization terms.

IV. NETWORK REPRESENTATION LEARNING ENHANCED RECOMMENDATION ALGORITHM

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) traditional social-network-based recommendation algorithms use the coarse-grained trust values, i.e. binary trust values, to represent the degree of trust between users. The granularity of such binary trust values is too coarse to distinguish the different levels of trust among users. 2) only the observed trust relationships are considered, whereas the implicit trust relationships are often ignored in traditional social-network-based recommender systems. The observed trust relationships only capture the local structure of social network, but implicit trust relationships encode the global structure of social network. Many users are highly probable to have large trust degrees between one another because of their shared neighboring connections, although these users have not formed any direct trust links.

In this paper, the observed trust relation is defined as the first-order trust (i.e., explicit trust relationships), and the trust relation induced by the neighborhood structure is named as 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. We adopt the LINE model [7] to infer the user trust relationships that preserve both local and global information from the original social network, and propose a network representation learning enhanced recommendation algorithm. The social-network-based recommendation methods generally assume that similar users or trusted users share common preferences, and a user is willing to accept the recommendations from his/her similar or trusted users. Specifically, during the training of recommendation models, typical social-network-based recommendation methods make the latent feature vectors of users as similar as possible if there are trust relationships or social links between them. In other words, the similarity between users' latent feature vectors reflects whether there are trust relationships or social links between them. Hence, to some extent, similarity and trust have similar semantics in the community of recommender systems. Although the LINE model originally is used to infer implicit similarity relationships by considering the neighborhood structures of nodes in information networks, it also can be used to discover implicit trust relationships by exploiting the local and global trust structures in trust information networks. Hence, in our proposed recommendation method, we basically adopt the LINE model to infer the trust relationships among users. Furthermore, the experimental results evaluate the effectiveness of adopting the LINE model to infer the trust relationships among users.

The framework of network representation learning enhanced recommendation algorithm is showed in Figure 1, including learning embedded representations of users, computing the fine-grained trust values, matrix factorization with the fine-grained trust values, and rating prediction. In the following sections, we firstly present the process of learning users' embedded representations by utilizing the LINE model, and then explain the recommendation model and parameter learning process.

A. LEARNING EMBEDDED REPRESENTATIONS OF USERS

The LINE model [7] is an important representative of network representation learning technique, which simultaneously retains the local and global structures of the information network. The local structure is represented by the observable links, which captures the first-order similarity between vertices. Meanwhile, the global structure is determined by the shared neighborhood structure of the vertices, which captures the second-order similarity between the vertices. In our proposed recommendation model, we generally apply the LINE model to learn users' embedded representations of the social network. The detailed process is demonstrated as follows.

The joint probability distribution between a user trust pair (u, v), which is used to model the first-order trust between users, is defined as follows:

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

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

$$\widehat{p}_1(x_u, x_v) = \frac{w_{uv}}{W} \tag{3}$$

where $W = \sum_{(u,v)\in E} w_{uv}$, and w_{uv} is the weight of the edge (u, v). We minimize the KL-divergence between the joint probability distribution and the empirical probability distribution to preserve the first-order trust in social network, formally, as follows:

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

Implicit trust implies that two users with similar neighbors are highly probable to share a large degree of trust 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, i.e. 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:

$$p_2(x_v|x_u) = \frac{exp(y_v^{*^T} y_u)}{\sum_{k=1}^{|U|} exp(y_k^{*^T} y_u)} \tag{5}$$

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

$$\widehat{p}_2(x_v|x_u) = \frac{w_{uv}}{d_u} \tag{6}$$

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

To preserve the second-order trust in social network, the following objective function is obtained by utilizing the KL-divergence:

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

The 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 trusts, respectively. Then, the two low-dimensional representations are concatenated as one low-dimensional feature vector to simultaneously preserve the local and global structures of social network. In other words, each vertex x_u is represented as $y_u \in R^d$, where $d = d_1 + d_2$.

B. NETWORK REPRESENTATION LEARNING ENHANCED RECOMMENDATION ALGORITHM

After using the LINE model to learn users' embedded representations, which preserve the local and global structures of social network, we utilize the inner product of the presentations to compute the fine-grained trust among users, formally, as follows:

$$s_{uv} = \frac{y_u^T y_v}{||y_u||_2 ||y_v||_2} \tag{8}$$

where y_u and y_v represent the low-dimensional feature representations of users u and v, respectively. The denominator is used to normalize s_{uv}. Compared with the coarse-grained trust value t_{uv}, it should be noted that the fine-grained trust value s_{uv} is more informative, and can accurately distinguish the different degrees 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 structures of the social network. In particular, even if there is no explicit connections between users, the implicit trust relationships between them can be deduced from their neighborhood structures.

In real life, users often have different preferences for different items. Meanwhile, users can be easily influenced by their friend community, and likely to accept their friends' recommendations. Similar to RSTE [4], we assume that the final rating of user u for item i is a trade-off between the user's own preference and his/her friends' preferences, and integrate them by the ensemble parameter α, i.e. the prediction rating of user u for item i is defined as:

$$\widehat{r}_{ui} = \alpha u_u^T v_i + (1 - \alpha) \sum_{w\in S(u)} s_{uw} u_w^T v_i \tag{9}$$

where $S(u)$ is the set of most trust neighbors of user u. The first item refers to user u's prediction rating for item i based on his/her own preference, while the second item refers to the prediction rating based on the preferences of his/her friends, and α is the weight parameter.

In addition, without loss of generality, we map the ratings r_{ui} 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 the logistic function $g(x) = 1/(1 + e^{-x})$ to limit the predicted ratings $\widehat{r}_{ui}$ within the range of [0,1]. Minimizing the sum-of-squared-error loss function as well as using the regularization terms to prevent overfitting, the objective function of the network representation learning enhanced recommendation algorithm is formalized as:

$$L = \frac{1}{2}\sum_{u=1}^{N}\sum_{i=1}^{M} I_{ui}^{R}(r_{ui} - g(\alpha u_u^T v_i + (1-\alpha) \times \sum_{w\in S(u)} s_{uw} u_w^T v_i))^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_V}{2}||V||_F^2 \quad (10)$$

where $S(u) = \{w|s_{uw} \geq \delta\}$ is the set of most trust neighbors of user u, and the parameter δ is the threshold of the trust value.

We adopt SGD to solve the local minimum solution of L, and learn the latent feature vectors u_u and v_i. The partial derivatives of the objective function L with respect to u_u and v_i are computed as:

$$\begin{aligned}\frac{\partial L}{\partial u_u} &= \alpha \sum_{i=1}^{M} I_{ui}^{R} g'(\alpha u_u^T v_i + (1-\alpha)\sum_{w\in S(u)} s_{uw} u_w^T v_i) v_i \\ &\quad \times (g(\alpha u_u^T v_i + (1-\alpha)\sum_{w\in S(u)} s_{uw} u_w^T v_i) - r_{ui}) \\ &\quad + (1-\alpha)\sum_{p\in S(u)}\sum_{i=1}^{M} I_{pi}^{R} g'(\alpha u_p^T v_i + (1-\alpha)\sum_{q\in S(p)} s_{pq} u_q^T v_i) \\ &\quad \times (g(\alpha u_p^T v_i + (1-\alpha)\sum_{q\in S(p)} s_{pq} u_q^T v_i) - r_{pi}) s_{pu} v_i + \lambda_U u_u\end{aligned} \quad (11)$$

$$\begin{aligned}\frac{\partial L}{\partial v_i} &= \sum_{u=1}^{N} I_{ui}^{R} g'(\alpha u_u^T v_i + (1-\alpha)\sum_{w\in S(u)} s_{uw} u_w^T v_i) \\ &\quad \times (g(\alpha u_u^T v_i + (1-\alpha)\sum_{w\in S(u)} s_{uw} u_w^T v_i) - r_{ui}) \\ &\quad \times (\alpha u_u + (1-\alpha)\sum_{w\in S(u)} s_{uw} u_w) + \lambda_V v_i\end{aligned} \quad (12)$$

where $g'(x) = e^{-x}/(1+e^{-x})^2$ is the derivative of the logistic function $g(x)$.

In our proposed recommendation approach, the main computation cost involves two parts: learning the embedded representations of users by adopting the LINE model and learning latent user and item feature vectors by integrating the fine-grained trust into the matrix factorization model. The computational complexity of learning the embedded representations of users is $O(d.n.|E|)$, where d is the dimension of embedded representations of users, while n is the number of negative samples drawn by the network representation learning model, and $|E|$ denotes the number of edges in social network. Therefore, the computational complexity of learning the embedded representations of users is linear with respect to the number of edges $|E|$. In addition, since the process of learning the embedded representations of users is offline, it does not lead to additional computation cost to the process of learning latent user and item feature vectors. The main computation cost of learning latent user and item feature vectors is to evaluate the objective function L and its gradients with respect to latent user and item feature vectors. The computational complexity of evaluating the objective function L is $O(\phi_R.K + \phi_R.\bar{t}.K)$, where ϕ_R is the number of nonzero entries in the user-item rating matrix R, and $\bar{t}$ indicates the average number of the most trusted neighbors of users. Since the user-item rating matrix R is extremely sparse, the value of ϕ_R is relatively small. In addition, we use the trust threshold δ to filter out most of the weak trust relationships, which indicates that the value of $\bar{t}$ is also relatively small. The time complexities of computing $\frac{\partial L}{\partial u_u}$ and $\frac{\partial L}{\partial v_i}$ are $O(\phi_R.\bar{t}.K + \phi_R.\bar{t}^2.K)$ and $O(\phi_R.K + \phi_R.\bar{t}.K)$, respectively. Hence, the total time complexity of learning latent user and item feature vectors in one iteration is $O(\phi_R.\bar{t}.K + \phi_R.\bar{t}^2.K)$, which indicates that our proposed recommendation model is able to scale to large datasets.

V. EMPIRICAL ANALYSIS

In this section, we conduct several experiments on real-world datasets to compare the performance of our proposed recommendation algorithm with other state-of-the-art methods. In addition, our proposed recommendation algorithm is denoted as "NPL_Rec".

A. DATASET

We choose Epinions and FilmTrust datasets to evaluate the performance of our proposed method. The FilmTrust dataset used in our experiments is provided by the study of [33]. It contains 35497 ratings, 1642 users, 2071 items, and 1853 trust relationships. The sparse level of the user-item rating matrix is 98.86%. The Epinions dataset employed is provided by the work of [34]. It contains 922267 ratings, 22166 users, 296277 items, and 355813 trust relationships. The sparse level of the user-item rating matrix is 99.986%.

B. EVALUATION METRIC

We use the root mean square error (*RMSE*), which is widely used evaluation metric in recommender systems, to evaluate the performance of recommendation algorithms. *RMSE* is defined as:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_{test}} |r_{ui} - \widehat{r}_{ui}|^2}{|R_{test}|}}, \quad (13)$$

where r_{ui} and $\widehat{r}_{ui}$ represent the actual and the predicted ratings, 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 [15] was proposed by Mnih and Salakhutdinov. PMF is 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/her friends' preferences.
- SocialMF: SocialMF [5] integrates a trust propagation mechanism into PMF to improve the accuracy of the recommendation algorithm.
- TrustMF: TrustMF [6] performs matrix factorization on the user trust matrix to map users into two different latent feature spaces, i.e. 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 studies or based on our experiments. Under the following 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 NPL_Rec, $\lambda_U = \lambda_V = 0.001$, $\alpha = 0.3$. It should be noted that, for the classic social-network-based recommendation models such as RSTE and SocialMF, we utilize all original social relationships contained in Epinions and FilmTrust to train the recommendation models, while for NPL_Rec, we employ inferred user trust relationships based on network representation learning described in Section IV-A to train the recommendation model.

TABLE 1. Performance comparison on Epinions.

Recommendation Algorithm	$RMSE(K=10)$	$RMSE(K=20)$
PMF	2.054174	2.138694
SoRec	1.102938	1.230585
RSTE	1.240436	1.409444
SocialMF	1.082406	1.221720
TrustMF	1.364107	1.534906
NPL_Rec	1.053627	1.055917

D. PERFORMANCE COMPARISON

We set $\delta = 0.95$ on Epinions and $\delta = 0.5$ on FilmTrust, respectively. And the dimensions of embedded presentation are $d = 256$ and $d = 128$ on Epinions and FilmTrust, respectively. Meanwhile, we evaluate all comparison methods with the dimension of latent feature vector $K = 10$ and $K = 20$. The experimental results of all comparison algorithms on the two datasets are shown in Tables 1 and 2.

TABLE 2. Performance comparison on FilmTrust.

Recommendation Algorithm	$RMSE(K=10)$	$RMSE(K=20)$
PMF	1.062290	1.157162
SoRec	0.842222	0.856792
RSTE	0.878196	0.882022
SocialMF	0.831957	0.858315
TrustMF	1.165906	1.338692
NPL_Rec	0.818756	0.801275

As shown in Tables 1 and 2, for Epinions dataset, PMF has the worst performance among all comparison algorithms, and all the social-network-based recommendation algorithms outperform PMF; for FilmTrust dataset, PMF outperforms TrustMF, and is inferior to the other social-network-based recommendation algorithms. Generally, this observation indicates that utilizing social network information can effectively improve the performance of the traditional collaborative filtering algorithm. Among the traditional social-network-based recommendation algorithms (namely SoRec, RSTE, SocialMF, TrsutMF), the performance of SocialMF is the best, showing that integrating trust propagation mechanism into the matrix factorization model is superior to the other recommendation models. On the two datasets, our proposed approach outperforms all other comparison algorithms, ascertaining the effectiveness of our proposed algorithm. On the Epinions and FilmTrust datasets, when $K = 20$, compared with the optimal results among PMF, SoRec, RSTE, SocialMF and TrustMF, the improvements of our proposed algorithm are 13.6% and 6.5%, respectively.

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. Specifically, a large δ means that the proposed recommendation model filters out the weak user trust relationships, and integrates strong trust relationships. On the contrary, a small δ means that the proposed recommendation model integrates relatively weak trust relationships. In this section, we perform a set of experiments to investigate the impact of the parameter δ on recommendation performance. On Epinions, we set δ to be 0.75, 0.8, 0.85, 0.9 and 0.95, and the dimension of embedded presentation $d = 128$. On FilmTrust, we set δ to be 0.4, 0.5, 0.6, 0.7 and 0.8, and the dimension of embedded presentation $d = 16$. In addition, we set the dimension of latent feature vectors $K = 10$ on Epinions and $K = 20$ on FilmTrust, respectively. The experimental results are presented in Figure 2.

As illustrated in Figure 2, the parameter δ does affect the performance of NPL_Rec. On the two datasets, the values of *RMSE* show similar trends: with the increases of δ, *RMSE* gradually decreases and recommendation accuracy increases, indicating that integrating strong fine-grained trust relationships is more beneficial to improve the performance of the NPL_Rec model. In addition, on the two datasets, our proposed recommendation algorithm does not achieve the lowest *RMSE* under the same δ. A possible explanation is

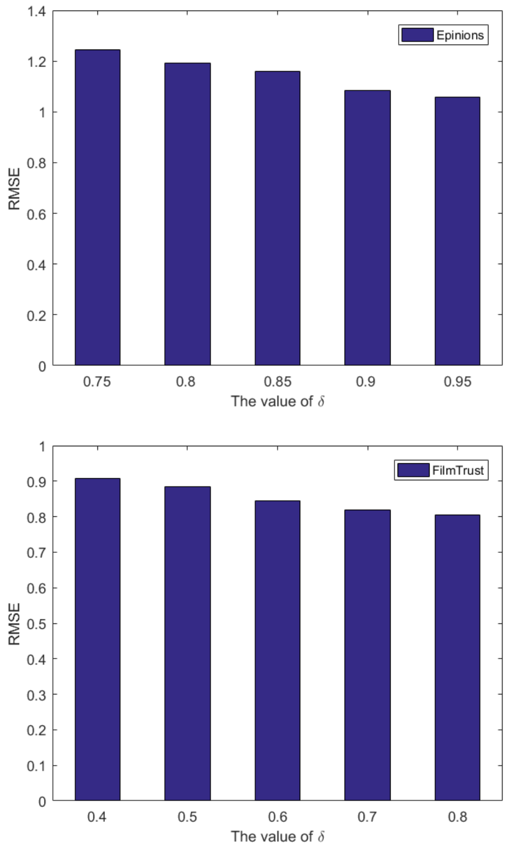


FIGURE 2. Impact of parameter δ.

that the network representation learning model used in our proposed recommendation algorithm infers different scales of trust values between users for different social networks. For example, the maximum trust value between users is 0.8839 on FilmTrust, while it is 0.9753 on Epinions.

F. IMPACT OF THE DIMENSION OF EMBEDDED REPRESENTATION

In this section, we vary the value of d, and investigate the impact of parameter d on recommendation quality. On Epinions, we set $\delta = 0.95$, $K = 10$, and vary d from 32 to 256. On Filmtrust, we set $K = 20$, and vary d from 16 to 256. In addition, since the FilmTrust dataset is small, the maximum trust between users rapidly decreases when increasing d. Hence, we set different δ values under different d. With different settings of d and δ, the best performances of NPL_Rec are reported. The experimental results are shown in Tables 3 and 4.

TABLE 3. The impact of d on Epinions.

The value of d	$RMSE(\delta = 0.95)$
d=32	1.162194
d=64	1.082012
d=128	1.057590
d=256	1.053627

TABLE 4. The impact of d on FilmTrust.

The value of d	$RMSE$
d=16	0.803757($\delta = 0.8$)
d=32	0.802497($\delta = 0.7$)
d=64	0.804478($\delta = 0.5$)
d=128	0.801275($\delta = 0.5$)
d=256	0.802624($\delta = 0.5$)

As illustrated in Tables 3 and 4, NPL_Rec obtains the best performance when $d = 256$ on Epinions. On FilmTrust, when $d = 128$, NPL_Rec achieves the best performance. In addition, with the increasing of d, it becomes more and more difficult to infer strong trust relationships between users. This is because the LINE model may encode more low-dimensional features, with the increasing of d, but it will also introduce some noise into the embedded representations of users, which negatively affects the accuracy of fined-grained trust values based the embedded representations. On two datasets, our proposed recommendation algorithm achieves the lowest *RMSE* under different settings of d. This indicates that, in order to accurately infer the fined-grained trust relationships, it is necessary for us to tune the value of d for different social networks.

G. IMPACT OF PARAMETER K

In this section, we fine-tune the value of K from 5 to 50, and observe the changing trends of *RMSE* on the two datasets. On Epinions, we set $\delta = 0.95$, $d = 256$; on FilmTrust, we set $\delta = 0.5$, $d = 128$. The experimental results are shown in Figure 3.

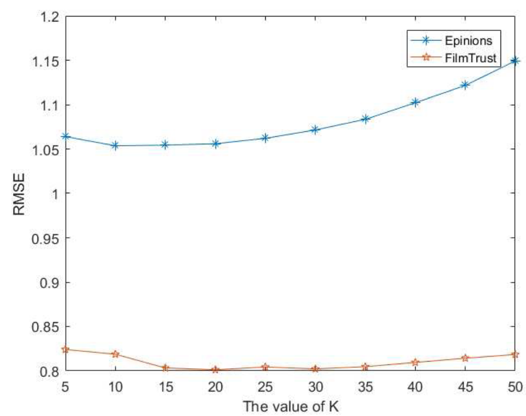


FIGURE 3. Impact of parameter K.

As shown in Figure 3, the recommendation quality of our proposed recommendation method is sensitive to the value of K. The recommendation quality firstly improves as K increases, and then degrades as the value of K further increases. Hence, a relatively large dimension of the latent feature vector is not beneficial for improving the recommendation performance. This observation confirms the assumption of matrix factorization that: only a small number of latent factors contribute to users' preferences and items'characteristics. NPL_Rec achieves the best performance when K is around 10 and 20 on Eipinions and FilmTrust, respectively.

H. THE FIRST-ORDER TRUST VERSUS THE SECOND-ORDER TRUST

In the process of learning users' low-dimensional representations from social networks using the LINE model, we adopt two different objective functions, i.e. O_1 and O_1, to learn two low-dimensional representations for each users, which capture the first-order and the second-order trusts, respectively. Then, the two low-dimensional representations are merged as one low-dimensional representation to encode both the first-order and the second-order trusts. In this section, we conduct a group of experiments to evaluate the effectiveness of this strategy. We denote our proposed network representation learning enhanced recommendation method with only using the first-order trust as NPL_Rec_FT, and another one that preserving the second-order trust as NPL_Rec_ST. On Epinions, we set $K = 10$, $d = 256$ for NPL_Rec_FT, NPL_Rec_ST and NPL_Rec. In particular, δ is 0.5 for NPL_Rec_FT since the first-order trust derived from Epinions is relatively small, which is described in Section V-E. And we assign $\delta = 0.95$ for NPL_Rec_ST and NPL_Rec. For FilmTrust, we set $\delta = 0.5$, $K = 20$, $d = 128$ for all comparison methods. The experimental results are plotted in Figure 4.

As shown in Figure 4, on both datasets, NPL_Rec consistently outperforms NPL_Rec_FT and NPL_Rec_ST. This observation demonstrates the effectiveness of our proposed recommendation method, which integrates both the first-order and the second-order trusts into the matrix factorization model. Moreover, NPL_Rec_ST is superior to NPL_Rec_FT, which shows that preserving the second-order trust is more beneficial for recommendation model than preserving the first-order trust.

I. EFFICIENCY COMPARISON

In this section, we compare the runtime of model training of our proposed method with that of other baselines to evaluate the efficiency of our proposed model. On Epinions, we set $K = 10$, $d = 256$, $\delta = 0.95$; on FilmTrust, we set $K = 20$, $d = 128$, $\delta = 0.5$. The parameter settings of other comparison algorithms are the same as those provided in Section V-C. The experimental results are presented in Table 5.

TABLE 5. The runtime of model training (hour : minute : second).

Recommendation Algorithm	Epinions	FilmTrust
PMF	00:00:10	00:00:01
SoRec	00:01:46	00:00:02
RSTE	04:10:37	00:00:16
SocialMF	00:51:23	00:00:03
TrustMF	00:03:49	00:00:07
NPL_Rec	00:01:06	00:00:02

As shown in Table 5, the runtime of training PMF is minimum. This is owing to the fact that PMF model only utilizes ratings to learn latent user and item feature vectors, and ignores social network information. Although RSTE and our proposed algorithms adopt the similar scheme that uses social network information to constrain users' latent feature vectors, RSTE needs more time for model training. In terms of the runtime of training model, our proposed algorithm is superior to SoRec, RSTE, SocialMF and TrustMF on Epinions; on FilmTrust, our proposed algorithm outperforms RSTE, SocialMF and TrustMF, and is comparable to SoRec.

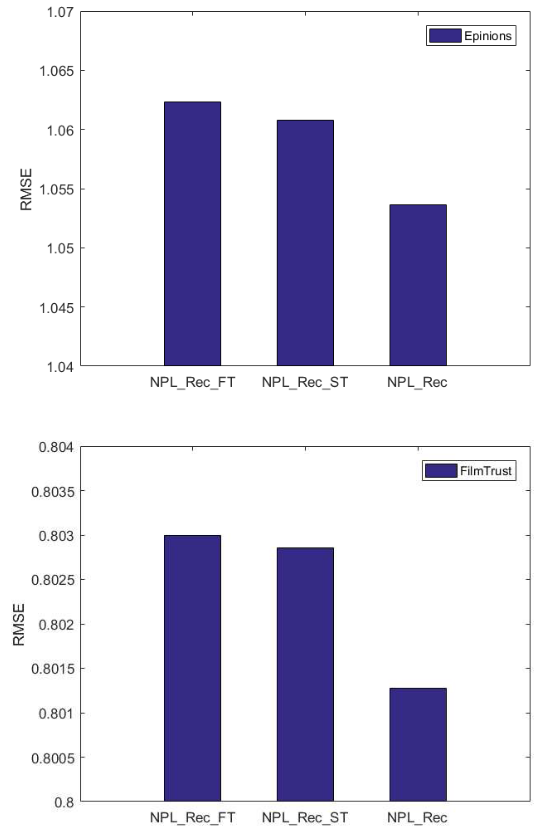


FIGURE 4. Impact of different strategies of inferring trust relationships.

J. IMPACT OF PARAMETER α

In our proposed recommendation model, the rating decision making is a trade-off between users' own preferences and friends' preferences, which are integrated together by the ensemble parameter α. Generally, the contributions of users' own preferences and friends' preferences to the rating decision making are balanced by α. In this section, we vary the value of α to investigate the sensitivity of the recommendation performance of NPL_Rec to the ensemble parameter α. On Epinions, we set $K = 10$, $\delta = 0.95$, $d = 256$. And on FilmTrust, we set $K = 20$, $\delta = 0.5$, $d = 128$. The experimental results are illustrated in Figure 5.

As shown in Figure 5, we observe that as α increases, *RMSE* firstly drops down quickly, and then begins to slowly move upwards when α surpasses a certain threshold. Meanwhile, we can observe that heavily depending on

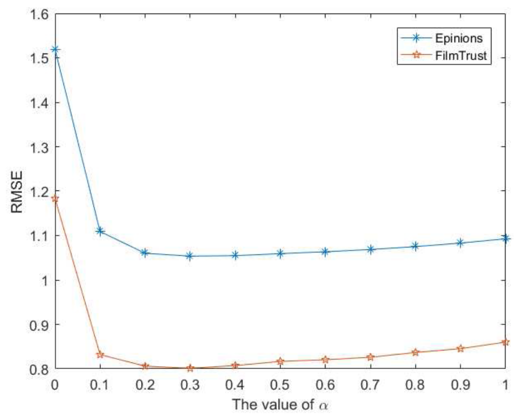


FIGURE 5. Impact of parameter α.

friends' preferences or completely ignoring them will degrade the recommendation performance of NPL_Rec. On both datasets, our proposed recommendation algorithm achieves the best performance when α is around 0.3. This implies that the final rating decision making of our proposed approach is more dependent on the social network information.

In short, according to the above empirical experimental results, the network representation learning enhanced recommendation algorithm proposed in this research shows great superiority over other comparison algorithms in terms of recommendation quality on the two real-world datasets. In terms of the efficiency, our proposed method is also comparable to other state-of-the-art social-network-based recommendation algorithms.

K. COMPARISON OF DIFFERENT NETWORK REPRESENTATION LEARNING SCHEMES

Besides the LINE model, both DeepWalk and node2vec models can also be used to learn the embedded representations of users from social networks, and the node2vec model is superior to the DeepWalk model. In this section, in order to justify the choice of adopting the LINE model to learn the embedded representations of users, we conduct another group of experiments to compare NLP_Rec against its variant by using the node2vec model rather than the LINE model to learn the embedded representations. We refer to the variant of NLP_Rec as NLP_Rec_Node2vec. For both NLP_Rec and NLP_Rec_Node2vec, we set $K = 20, \delta = 0.95, d = 256$ on Epinions, while on FilmTrust, we set $K = 10, \delta = 0.5$, $d = 128$ for NLP_Rec, and $K = 10, \delta = 0.7, d = 32$ for NLP_Rec_Node2vec. In addition, we set the bias parameters $p = q = 0.25$ for node2vec. Under these parameter settings, NLP_Rec and NLP_Rec_Node2vec achieve their optimal performance. The experimental results are presented in Table 6.

As shown in Table 6, NLP_Rec is superior to NLP_Rec_Node2vec on both datasets, which justifies the choice of adopting the LINE model to learn the embedded representations. This observation implies that the LINE model is more effective than node2vec in simultaneously capturing the first-order and second-order trust relationships.

TABLE 6. Performance comparison of different network representation learning models for recommendation.

Recommendation Algorithm	Epinions	FilmTrust
NLP_Rec	1.053627	0.801275
NLP_Rec_N	1.056102	0.916278

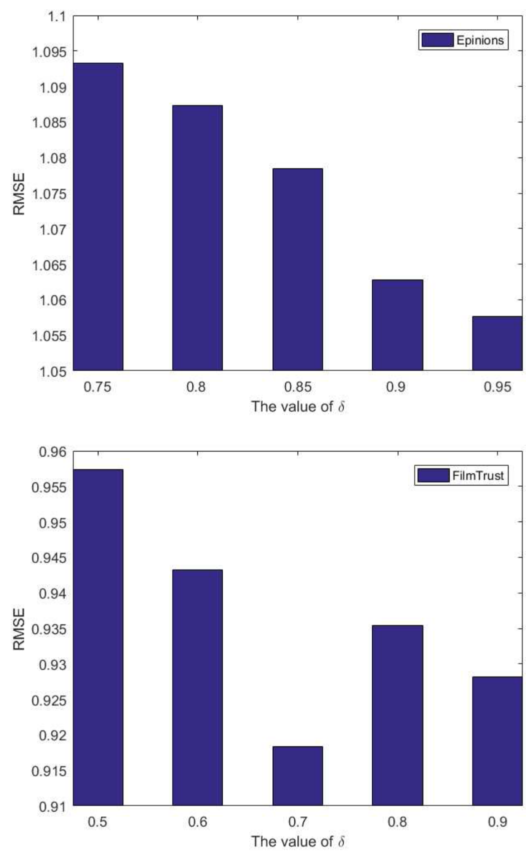


FIGURE 6. Impact of parameter δ on NLP_Rec_Node2vec.

Moreover, we conduct a group of experiments to investigate the impact of parameter δ on the recommendation performance of NLP_Rec_Node2vec. We set the bias parameters $p = q = 0.25$ for node2vec, and other parameter settings of NLP_Rec_Node2vec are same as the settings of NLP_Rec in Section V-E. The experimental results are illustrated in Figure 6. As indicated in Figure 6, NLP_Rec_Node2vec is sensitive to the value of δ. On Epinions and FilmTrust datasets, NLP_Rec_Node2vec achieves its best performance when δ is around 0.95 and 0.7, respectively.

Furthermore, we also conduct another group of experiments to investigate the impact of d on recommendation quality of NLP_Rec_Node2vec. On Epinions and FilmTrust, we set $\delta = 0.95$ and $\delta = 0.7$, respectively. Other parameters remain the same. The experimental results are plotted in Figure 7. As shown in Figure 7, the value of d also significantly affects the performance of NLP_Rec_Node2vec.

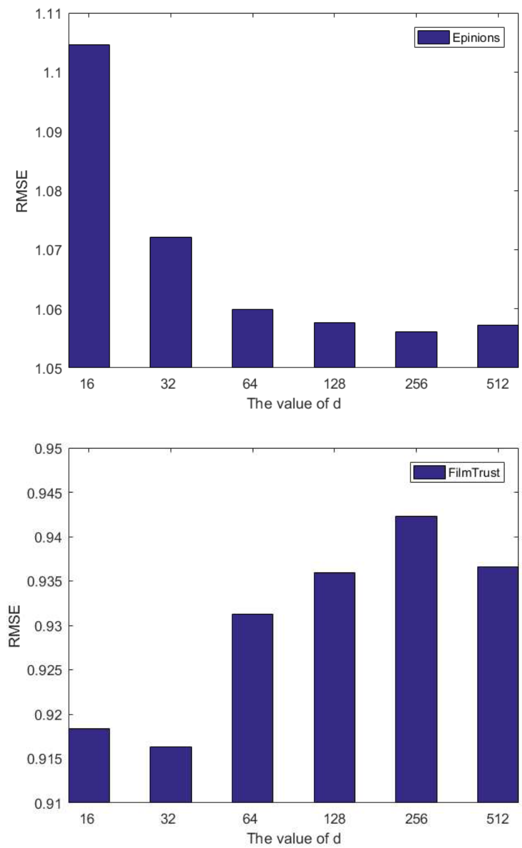


FIGURE 7. Impact of parameter *d* on NLP_Rec_Node2vec.

As d increases, the value of RMSE firstly drops down. After d reaches a certain threshold, the RMSE begins to increase as d increases, which indicates that the performance degrades when d is too large. This is also owing to the fact that the network representation learning model with a relatively large d may introduce some noise into the embedded representations of users, which affects the computing of the fine-grained trust values.

VI. CONCLUSION

Traditional social-network-based recommendation algorithms generally utilize the coarse-grained trust relationships to generate recommendations, which seriously hinders the performance of recommendation algorithms. To tackle this problem, we proposed a network representation learning enhanced recommendation algorithm in this study. Specifically, we first adopt a network representation learning technique to embed a social network into a low-dimensional space, and then utilize the low-dimensional representations of users to infer fine-grained dense trust relationships between them. Finally, we integrate the fine-grained dense trust relationships into the classic 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.

As mentioned above, our proposed recommendation algorithm is a two-stage approach, i.e. firstly adopting a network representation technique to embed a social network into a low-dimensional space, and then integrating the fine-grained dense trust relationships inferred from embedded representations of users into the matrix factorization model. In this two-stage learning model, users have different low-dimensional representations in the social network as well as rating information, which may lead to the semantic gap between the social network structure and ratings. In future work, we aim to explore how to integrate network representation learning and the matrix factorization technique to learn unified feature representations of users to further enhance performance of our proposed recommendation model.

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," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[2] J. S. Breese, D. Heckerman, and C. Kadie, *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Burlington, MA, USA: Morgan Kaufmann, 1998, p. 18.

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

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

[5] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. RecSys*, 2010, pp. 135–142.

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

[7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. World Wide Web Conf. Steering Committee*, 2015, pp. 1067–1077

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. CSCW*, 1994, pp. 175–186.

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

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

[11] G.-R. Xue *et al.*, "Scalable collaborative filtering using cluster-based smoothing," in *Proc. SIGIR*, 2005, pp. 114–121.

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

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

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

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

[16] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

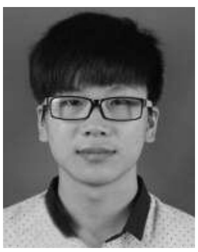
[17] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. KDD*, 2008, pp. 426–434.

[18] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 713–719.

[19] 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.

[20] H. Cai, V. W. Zheng, and K. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

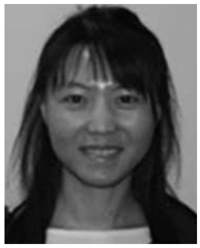
[21] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowl.-Based Syst.*, vol. 151, pp. 78–94, Jul. 2018.
[22] S. Bhagat, G. Cormode, and S. Muthukrishnan, *Node Classification in Social Networks*. Boston, MA, USA: Springer, 2011, pp. 115–148.
[23] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[24] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *Proc. 12th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2003, pp. 556–559.
[25] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proc. WWW*, 2013, pp. 37–48
[26] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. KDD*, 2014, pp. 701–710.
[27] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. KDD*, 2016, pp. 855–864.
[28] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, PTE, and node2vec," in *Proc. WSDM*, 2018, pp. 459–467.
[29] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. KDD*, 2015, pp. 1165–1174.
[30] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. KDD*, 2017, pp. 135–144.
[31] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. KDD*, 2018, pp. 1177–1186.
[32] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based POI embedding for location-based recommendation," in *Proc. CIKM*, 2016, pp. 15–24.
[33] J. Golbeck and J. Hendler, "FilmTrust: Movie recommendations using trust in Web-based social networks," in *Proc. 3rd IEEE Consum. Commun. Netw. Conf. (CCNC)*, vol. 1, Jan. 2006, pp. 282–286.
[34] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2712–2718.



QIANG WANG is currently pursuing the bachelor's degree with Tongda College, Nanjing University of Posts and Telecommunications. His main research interests include machine learning and recommender systems.

YONGHONG YU received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, and the Ph.D. degree in computer application technology from Nanjing University. He is currently an Associate Professor with Tongda College, Nanjing University of Posts and Telecommunications. His main research interests include machine learning, recommender systems, and image processing.

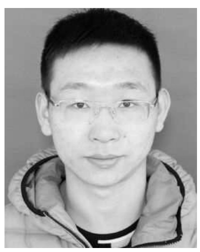
HAIYAN GAO received the M.Sc. degree from the Nanjing University of Aeronautics and Astronautics, China. She is currently a Lecturer with the Tongda College, Nanjing University of Posts and Telecommunications. Her main research interests include data mining and user behavior analysis.



LI ZHANG received the Ph.D. degree from the University of Birmingham. She is currently an Associate Professor and a Reader in computer science with Northumbria University, U.K., and also serving as an Honorary Research Fellow with the University of Birmingham, U.K. She is expertise in artificial intelligence, machine learning, evolutionary computation, and deep learning. She has served as an Associate Editor for *Decision Support Systems*.

YANG CAO received the B.Sc. and M.Sc. degrees from Anhui Normal University, China. He is currently an Assistant Researcher with the Tongda College, Nanjing University of Posts and Telecommunications. His main research interests include data mining and deep learning.

LIN MAO is currently pursuing the bachelor's degree with the Tongda College, Nanjing University of Posts and Telecommunications. His main research interest is the recommendation algorithms.



KAIQI DOU is currently pursuing the bachelor's degree with the Tongda College, Nanjing University of Posts and Telecommunications. His main research interests include recommendation algorithms and image processing.

WENYE NI is currently pursuing the bachelor's degree with Tongda College, Nanjing University of Posts and Telecommunications. Her main research interests include recommendation algorithms and deep learning.