Guo L, Wen YF, Wang XH. Exploiting pre-trained network embeddings for recommendations in social networks. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(4): 682–696 July 2018. DOI 10.1007/s11390-018-1849-9

Exploiting Pre-Trained Network Embeddings for Recommendations in Social Networks

Lei Guo[1,2], *Member*, *CCF*, Yu-Fei Wen[2], and Xin-Hua Wang[3]

[1] *Postdoctoral Research Station of Management Science and Engineering, Shandong Normal University Jinan 250014, China*

[2] *School of Management Science and Engineering, Shandong Normal University, Jinan 250014, China*

[3] *School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China*

E-mail: leiguo.cs@gmail.com; wenyufei92@sina.com; wangxinhua@sdnu.edu.cn

Received January 15, 2018; revised May 18, 2018.

Abstract Recommender systems as one of the most efficient information filtering techniques have been widely studied in recent years. However, traditional recommender systems only utilize user-item rating matrix for recommendations, and the social connections and item sequential patterns are ignored. But in our real life, we always turn to our friends for recommendations, and often select the items that have similar sequential patterns. In order to overcome these challenges, many studies have taken social connections and sequential information into account to enhance recommender systems. Although these existing studies have achieved good results, most of them regard social influence and sequential information as regularization terms, and the deep structure hidden in social networks and rating patterns has not been fully explored. On the other hand, neural network based embedding methods have shown their power in many recommendation tasks with their ability to extract high-level representations from raw data. Motivated by the above observations, we take the advantage of network embedding techniques and propose an embedding-based recommendation method, which is composed of the embedding model and the collaborative filtering model. Specifically, to exploit the deep structure hidden in social networks and rating patterns, a neural network based embedding model is first pre-trained, where the external user and item representations are extracted. Then, we incorporate these extracted factors into a collaborative filtering model by fusing them with latent factors linearly, where our method not only can leverage the external information to enhance recommendation, but also can exploit the advantage of collaborative filtering techniques. Experimental results on two real-world datasets demonstrate the effectiveness of our proposed method and the importance of these external extracted factors.

Keywords social recommendation, network embedding, matrix factorization, item sequential pattern

1 Introduction

With the development of computer technology, the Internet provides a convenient way for users to communicate with each other while it also faces the information overload problem. How to help users find useful information from vast amounts of user-generated data has become one of the most urgent problems to be solved. The recommender system as one of the effective information filtering technologies has attracted lots of attention in recent years, which tries to solve the information overload problem by suggesting items (e.g., locations, books, and movies) to the users that have potential interest to them. Typical recommender systems are collaborative filtering based matrix factorization (MF) models, which try to learn the latent representations of users and items by leveraging their similar neighbors in a matrix completion task[1-2].

However, traditional methods only utilize the user-item rating matrix for recommendation, and the social connections (e.g., friends, trust relations) and item sequential patterns are ignored. But in our real life, we

Regular Paper

Special Section on Recommender Systems with Big Data

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61602282, 61772321, 61472231 and 71301086, and the China Postdoctoral Science Foundation under Grant No. 2016M602181.

always turn to our friends that we trust for recommendations, and often select the items that have similar sequential patterns. Social connections and sequential patterns are important factors for us to recommend the right items. Taking the recommendation scenario in social network as an example, suppose we have two users (u_1 and u_2) and two items (v_1 and v_2), where u_1 and u_2 trust each other. In this scenario, when we recommend items to u_1, we will also consider the items that have been preferred by u_2. Moreover, if we know items v_1 and v_2 are often rated sequentially by user u_2, we will have a high confidence to recommend v_2 to u_1, since he/she is very likely to rate v_2 next time as user u_2 has done. The social connections help us locate the items that we are potentially interested in. The item sequential patterns help us find out the items that are rated sequentially.

How to overcome these challenges to further improve the recommendation performance has become one of the hot spots in recent studies[3-5]. Ma *et al.*[5] treated social networks as the social constraints on recommender systems and designed a matrix factorization objective function with social regularization. Guo *et al.*[6] learned the user feature factors by ranking items correctly and proposed a category sensitive random walk method to explore the impact of social relations with different facets. Wang *et al.*[7] proposed a sequential personalized recommendation method for spatial items, which introduces the topic-region variable to model the sequential influence. Although these existing studies have achieved good results, most of them regard social influence and sequential information as regularization terms, and the deep structure hidden in social networks and rating patterns has not been fully explored.

On the other hand, deep learning based embedding technique has shown its power in many recommendation tasks with its capability of extracting hierarchical representations from raw data. Chu and Tsai[8] extracted the visual features from photos taken by customers by utilizing a convolutional neural network model, and then incorporated them into the collaborative filtering based recommendation methods. Liang *et al.*[9] trained a neural network on semantic tagging information as a content model and used it as a prior in a collaborative filtering model. The application of the embedding technique is not limited to images, contents and music, and it also provides an effective way to explore the network structure patterns by using low-dimensional embedding vectors[10]. Perozzi *et al.*[11] proposed the DeepWalk approach to learn the representations of vertices in a network, which uses local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences. Grover and Leskovec[12] extended the DeepWalk method and proposed a novel leaning method named node2vec by defining a flexible notion of a node's network neighborhood and designing a biased random walk procedure. But these studies focus on learning node representations by investigating the network structure deeply, and cannot directly learn the embeddings of item sequential patterns. Moreover, the performance of integrating external factors deduced by neural networks and latent factors learned by collaborative filtering techniques is not clear.

Motivated by the above observations, we take advantage of the network embedding techniques to learn the deep information hidden in social connections and rating patterns, and propose an embedding-based social recommendation method, which is composed of the embedding model and the collaborative filtering model. In the embedding model, a neural network, i.e., the node2vec model[12], is pre-trained to learn the user and the item representations from the corresponding social networks and the item sequential patterns, respectively. In the collaborative filtering model, the latent factor model matrix factorization is introduced to deal with the rating matrix, from which the low-rank latent factors are learned. To exploit the learned embeddings, the outputs of the node2vec are treated as the high-level representations of social and sequential context, and they are further fused with the latent factors by a linear model. To evaluate the effectiveness of our proposed method, we conduct experiments on two real-world social networks, and the results indicate that our method is more effective and can achieve a better performance than other related recommendation methods.

In summary, the main contributions of this paper are as follows.

- We utilize the network embedding model (node2vec) to learn the representations of users and items from the social network and a pre-defined item sequential network.
- We propose an embedding-based recommendation model MFn2v+ by integrating the learned embeddings from node2vec with the latent factors from the collaborative filtering model linearly, which can exploit the advantages of these two models simultaneously.
- We conduct experiments on two real-world datasets to evaluate the performance of our proposed

method and the influence of these external extracted factors.

The remainder of this paper is organized as follows. Section 2 first describes the recommendation problem we study in this work, and then briefly discusses the related recommendation methods. Section 3 reviews the basic low-rank matrix factorization method, which is introduced as the basic recommendation method of our work. Section 4 combines the embedding model with latent factor model, and proposes two embedding-based recommendation methods MFn2v and MFn2v+. Section 5 shows the comparison results of MFn2v+ with other related methods in two real-world datasets. Section 6 outlines some conclusions and directions for future work.

2 Preliminary

In this section, we first describe the recommendation scenario in social networks. Second, we review and discuss some related recommendation methods.

2.1 Problem Statement

Let us consider the recommendation problem in social networks, where users can express their preferences by rating on different items, and the more they like the item, the higher the rating they give. But different from traditional recommender systems, as we always turn to our friends for recommendations, users can make social connections to express their social interest by connecting with different users, and the more common friends they share, the more similar interest they have. In addition to the social relationships, users also express their interest by rating items in different sequence orders. The item sequential pattern provides another way for us to learn user model deeply. The recommendation scenario in social network is shown in Fig.1, which includes three central elements: the user-item rating matrix, social connections among users, and item sequential patterns.

Suppose we have a social network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ (as shown in Fig.1(a)), where $\mathcal{U}$ is the user set, and $\mathcal{E}$ is the social relation set. Each user $u \in \mathcal{U}$ makes some users that have similar interest be his/her friends. Except these social relationships, users also rate some items (as shown in Fig.1(b)) in different sequence orders to express their favors, where the observed rating data is denoted as real values (from 1 to 5), and the unknown data is denoted as "?" (as shown in Fig.1(c)). In this scenario, when we recommend items to users, we will consider not only the user's rating history, but also the social connections and the item sequential patterns. Hence, how to make effective recommendations by utilizing above contexts in social networks is the main task of this work.

2.2 Related Work

Typical recommender systems are collaborative filtering based techniques[13], which can be categorized into two classes: memory-based[14-15] and model-based approaches[16-18]. Compared with memory-based methods, which directly make predictions based on the strategies that can find similar users or items, model-based methods focus on employing machine learning and statistical techniques to learn a compact model to make predictions. For example, Hofmann[19] presented a powerful method for collaborative filtering based on a generalization of probabilistic latent semantic analysis to continuous-valued response variables, which is highly scalable and extremely flexible. Salakhutdinov and Mnih[2] presented the probabilistic matrix factorization (PMF) model to make predictions on the large, sparse

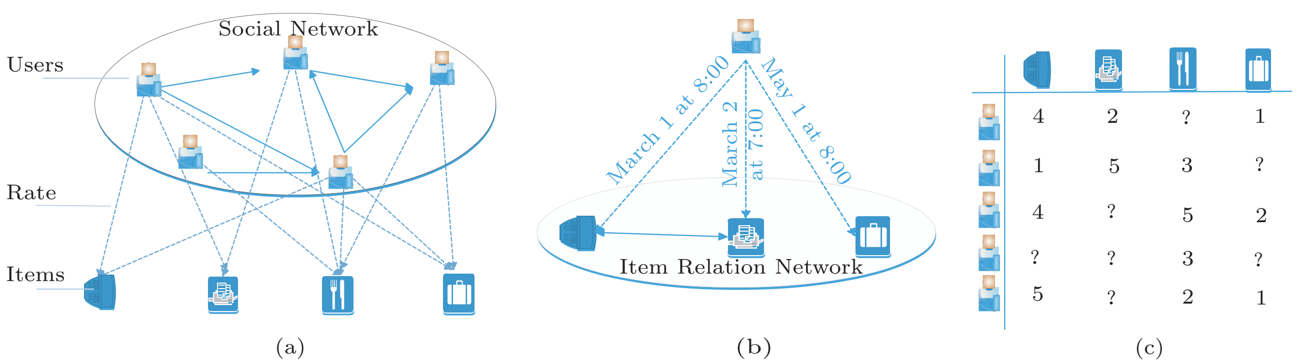


Fig.1. Example of the recommendation scenario in social network. (a) One typical example of the social network. (b) Example of item relation network. (c) User-item rating matrix.

and very imbalanced Netflix dataset, and achieved a better score than other related recommendation methods. However, these traditional recommendation methods only utilize rating history for recommendation, and the social connections and the item sequential patterns are ignored. To overcome these challenges, many researchers began to take social context information to enhance traditional recommender systems. For example, Menon and Elkan[20] introduced a log-linear model with latent features (LFL) for user-item pair prediction, where the side information can be easily exploited for predictions. In this work, the authors measured the interactions between the inputs and labels by feature functions to train the log-linear model, and then extended the model by adding latent features. One of the strengths of this model is that it can easily handle the side information existing in the system. According to the information that has been considered, existing studies can be classified as social network based recommendation methods and temporal-aware recommendations.

Social Network Based Recommendations. In real-world social networks, as users always seek advice from their friends, exploring the social connections among users for recommendation has been widely studied in recent years[21–25]. For example, to overcome the data sparsity of the input rating matrix, Massa and Avesani[21] exploited the trust information explicitly expressed by users to search for trustable users, and the items appreciated by these users are then recommended to the target user. Instead of simply considering rating from trusted neighbors for recommendation, Jamali and Ester[22] proposed a random walk model combining the trust-based and the collaborative filtering model for recommendation, which allows us to define and measure the confidence of a recommendation. However, the above existing methods are memory-based approaches, which are based on literal memory of past rating, and cannot be extended to very large datasets. Hence, some model-based approaches are further proposed to overcome this challenge[4,26–31]. For example, Ma *et al.*[4] investigated the social recommendation problem in a latent factor model, i.e., matrix factorization method, where the low-rank latent factors of users and items are learned. To incorporate the social connections for recommendation, a recommendation method that fuses the users' tastes and their trusted friends' favors together was proposed[32]. Jamali and Ester[32] incorporated the mechanism of trust propagation into the matrix factorization model and proposed a novel model-based approach named SocialMF, which assumes the factors of every user are dependent on the factor vectors of his/her direct neighbors in the social network. Then, the user latent factors are composed by two components, that is, the zero-mean Gaussian prior that avoids over-fitting and the conditional distribution of user latent features given the latent features of his/her direct neighbors. With this idea, latent factors of users indirectly connected in the social network will be dependent and hence the trusts get propagated. By performing gradient descent for all users and items, a local minimum can be found. This method has achieved the state-of-the-art recommendation performance in many public datasets and recommendation scenarios. Differing from [32], Ma *et al.*[33] interpreted the differences between social-based recommender systems and trust-aware recommender systems, and proposed the social regularization based recommendation method (SoReg) to further improve traditional recommender systems. In their work, they conducted the vector space similarity (VSS) and Pearson correlation coefficient (PCC) to measure the similarities of two users, and modeled the social network information as social regularization terms to constrain the matrix factorization objective function. In our experiments, the individual-based regularization model is utilized, which is sensitive to those users whose friends have diverse tastes. Motivated by the intuition that users are likely to seek suggestions from both their local friends and users with high global reputations, Tang *et al.*[34] exploited social relations from local and global perspectives for online recommender systems, where the global context is captured by weighting the importance of user rating according to users' reputation scores.

Temporal and Sequence-Aware Recommendations. Inspired by the intuition that successively rated items are more likely to be correlated, that is, the sequence of a user's behaviors within a specific time interval can reflect the user's rating interest, researchers have also exploited the temporal and item sequence context for recommendation[7,35–39]. For instance, Zhang and Chow[40] proposed a probabilistic framework for time-aware location recommendations, which estimates the time probability density of a user visiting a new location without splitting the continuous time into discrete time slots. To capture temporal dynamics, Liu *et al.*[37] extended the collaborative ranking model with a time-aware parameter and introduced a novel temporal smoothness regularization term to avoid overfitting. Zhang *et al.*[41] exploited sequential influence on location recommendations, where they first represented the mined sequential patterns from location sequences as a dynamic location-location transition Graph (L^2TG), and then predicted whether a user will visit a location by additive Markov chain. Finally, a unified recommendation framework that fuses sequential influence with geographical influence and social influence was proposed. Wang *et al.*[7] introduced a novel latent variable topic-region to model and fuse sequential influence with personal interest in the latent space and the exponential space, and a sequential personalized spatial item recommendation framework was proposed.

Except the social, temporal and item sequential contexts, other types of contexts have also been explored[42-47]. For example, Carrillo *et al.*[46] presented context-aware user profiles, in which the profile definitions are associated with particular situations encountered by the users. Lian *et al.*[47] advised to take the geographical influence into account as context and developed the model that joints geographical modeling and matrix factorization (GeoMF), which uses the augmented latent factors to model the users' clustering phenomenon. Zheng *et al.*[48] focused on how to take the effect of emotions in recommendations. Specifically, they explored the usage of emotions to discover how emotional features interact with those context-aware recommendation algorithms in the recommendation process.

However, most of these previous studies mainly consider social and item contexts as regularization terms, and most of them do not incorporate these contexts by utilizing the embedding techniques, which have achieved great success in many other related tasks.

Embedding-Based Recommendation Methods. With the development of deep neural networks[49-52], distributed representation methods and embedding models have been well studied in recent years[53-57]. For instance, Mikolov *et al.*[56] showed how to train distributed representations of words and phrases with the Skip-gram model. Tang *et al.*[57] proposed a novel network embedding method, which can easily scale up to networks with millions of vertices and billions of edges. As embedding methods have the capability of extracting hierarchical representations from raw data, many researchers have also tried to incorporate these extracted factors into recommender systems. For example, Zhao *et al.*[10] presented a novel perspective to address the recommendation task by utilizing the network representation learning techniques, which first transforms the adoption records into a k-partite adoption network, and then applies the network embedding approach to learn vertex embeddings. In this way, the recommendation task is casted into a similarity evaluation process using embedding vectors. Inspired by the success of word embedding models, Liang *et al.*[58] learned item embeddings using the sets of items each user has consumed, and further proposed a co-factorization model to jointly decompose the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors. To leverage the knowledge extracted from social networking sites for cross-site cold-start product recommendation, Zhao *et al.*[59] learned both users' and products' feature representations from e-commerce website using recurrent neural networks, and then developed a feature-based matrix factorization approach to leverage the learnt user embeddings for cold-start product recommendation.

Inspired by these existing studies, we leverage the factors hidden in the deep structures of social connections and item relations in an embedding-based recommender system. Our proposed method differs from the existing approaches in two aspects. First, we formalize the relationships of users and items into a predefined network respectively, and then extract the external representations by utilizing network embedding techniques. Second, we integrate the extracted network embeddings and the latent factors learnt from collaborative filtering models in a fused framework, which can benefit from these two models.

3 Low-Rank Matrix Factorization

To learn the latent factors from user rating data, we introduce the low-rank matrix factorization (LMF) method as our basic recommendation framework. The premise behind the LMF model[60] is that there is only a small number of factors influencing the users' preferences, and a user's preference vector is determined by how each factor applies to that user, which facilitates rapid dimensionality reduction of big data[2,61]. LMF model also offers much flexibility and enables us to integrate side information to make more accurate recommendations.

Suppose we have an $M \times N$ rating matrix $\boldsymbol{R} = (r_{u,i})_{M\times N}$ describing the numerical rating of M users on N items, where each entry $r_{u,i}$ denotes the rating of user u on item i. LMF approach seeks to approximate the rating matrix $\boldsymbol{R}$ by a multiplication of k-rank factors, and its objective function can be arrived as:

$$\mathcal{L}(\boldsymbol{U}, \boldsymbol{I}) = \min_{\boldsymbol{U}, \boldsymbol{I}} \frac{1}{2} \sum_{u=1}^{M} \sum_{i=1}^{N} (r_{u,i} - \boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i)^2 + \frac{\lambda_{\boldsymbol{U}}}{2} \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{I}}}{2} \|\boldsymbol{I}\|_{\mathrm{F}}^2, \quad (1)$$

where $\boldsymbol{U}$ and $\boldsymbol{I}$ are the latent feature factors of users and items, with column vectors $\boldsymbol{U}_u \in \mathbb{R}^k$ and $\boldsymbol{I}_i \in \mathbb{R}^k$ representing user-specific and item-specific feature vectors, respectively. $\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i$ is the predicted score for pair (u, i), and $\frac{\lambda_{\boldsymbol{U}}}{2} \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{I}}}{2} \|\boldsymbol{I}\|_{\mathrm{F}}^2$ is the corresponding regularization term. $|| \cdot ||_{\mathrm{F}}^2$ denotes the Frobenius norm. LMF minimizes the sum-of-squared-errors objective function with quadratic regularization terms, and gradient-based approaches can be applied to find a local minimum.

Although the LMF method has achieved great successes in many recommender systems, it assumes users and items are independent and identically distributed (i.i.d.), and only rating data is used in this method. The information from social connections and item sequences is ignored, which is important in social recommendation scenarios.

4 Combined with Network Embedding Factors

In this section, we first introduce the network embedding technique (node2vec) to pre-learn the external extracted factors. Then, we explore and discuss how to exploit user-trust relationship and item relationship to assist recommendation. In particular, we propose to partition rating dimensions into latent factors and network embedding factors (as shown in Fig.2), where the latent factors can be obtained by directly factorizing the rating matrix, and the network embedding factors can be achieved by pre-training the pre-defined user social network and item sequential network.

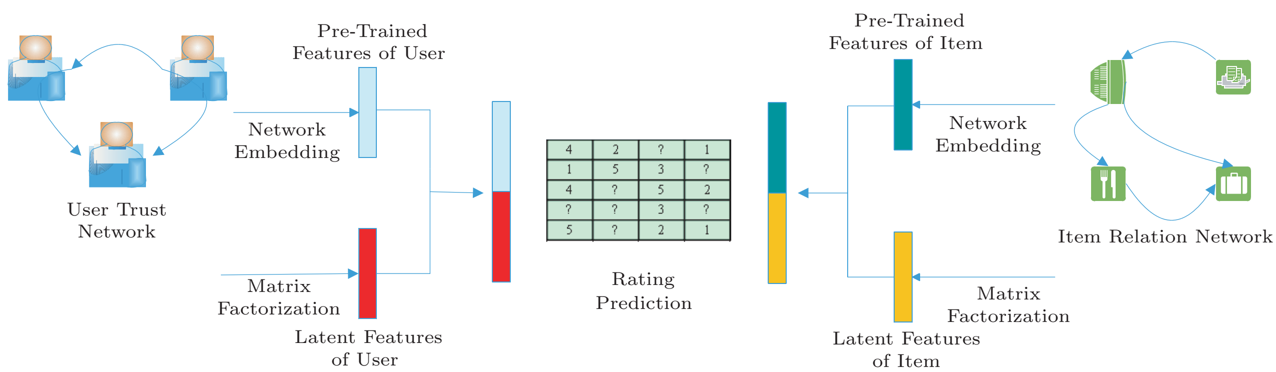


Fig.2. Diagram of our preference predictor, where rating dimensions consist of latent factors and external embedding factors.

4.1 node2vec Algorithm

Inspired by the success of neural network based embedding model in link prediction and node classification task, in this work we pre-train a network embedding model node2vec[1] in a semi-supervised task, and use the learnt embeddings as the representations of users and items. node2vec can learn the high-level continuous feature representations for the nodes within any given network, and can capture the diversity of connectivity patterns observed in networks by a biased random walk.

Let $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ be the given network, where $\mathcal{A}$ denotes the node set and $\mathcal{E}$ denotes the edge set. For every source node $m \in \mathcal{A}$, let $\mathcal{N}_S(m) \subset \mathcal{A}$ be the network neighborhoods of node m generated through a neighborhood sampling strategy S, that is, a biased random walk method that can explore neighborhoods in a breadth-first sampling as well as a depth-first sampling fashion. To learn the high-level representations of every node, node2vec tries to maximize the log-probability of observing a network neighborhood $N_S(m)$ for a node m conditioned on its feature representations:

$$\max_f \sum_{m \in \mathcal{A}} \log \Pr(N_S(m) | f(m)),$$

where f is the mapping function from nodes to low-dimensional feature representations. In our work, we let d represent the dimensions of feature representations, and then f can be formally expressed as a matrix of parameters with size $|\mathcal{A}| \times d$.

[1] https://snap.stanford.edu/node2vec/, May 2018.

In node2vec, p and q are two important parameters, which determine how fast the walk explores and leaves the neighborhood of the starting node. More specifically, parameter p controls how much the walk would like to immediately revisit a node. A high value (larger than $\max(q,\ 1)$) of p indicates that we are unlikely to sample an already existing node, whereas a low value (smaller than $\min(q,\ 1)$) of p would keep the walk close to the starting node. Parameter q allows the search to differentiate between "inward" and "outward" nodes, where a high value of q biases the walk to visit the nodes close to the starting node, and a low value of q biases the walk to visit the nodes far from the starting node.

To evaluate the quality of the pre-learned representations, we utilize an end-to-end method, that is, evaluating node2vec by how much our recommendation method can be improved. To find the appropriate parameter settings, we use a grid search over $p, q \in \{0.25, 0.50, 1, 24\}$ (as suggested by the author of node2vec) and choose the parameter values that make our method perform the best as our final setting (the specific setting can be seen in Subsection 4.2 and Subsection 4.3).

4.2 MFn2v Model

As users in social networks often express their social interest by making different friends, a better understanding of these social networks is potentially helpful for recommendation. A typical social network among users can be defined as follows.

Definition 1 (Social Network)[4]. *The social network is denoted by* $\mathcal{G}_{uu} = (\mathcal{U},\ \mathcal{E}_{uu})$*, which captures the social connections among users.* $\mathcal{U}$ *is the user set, and* $\mathcal{E}_{uu}$ *is the edge set. For any two users* $u_p \in \mathcal{U}$ *and* $u_m \in \mathcal{U}$*, if user* u_p *has a social connection with user* u_m *(such as trust relationship and friend relationship), there will be an edge* $e_{pm} \in \mathcal{E}_{uu}$ *from* u_p *to* u_m*; otherwise none.*

As the output of node2vec can be interpreted as the high-level representations of network nodes, we pretrain node2vec to mine the deep social structure of the given social network② $\mathcal{G}_{uu}$, and let $\boldsymbol{X}_u \in \mathbb{R}^d$ represent the extracted factors of user u from the given social network. These latent factors denote the deep social interest of users. A linear combination of them indicates to what extent a user will establish a social link with others. This information can be useful in predicting ratings, especially when a user has only rated very few items in history. Since the social network and the rating preference potentially encode different types of information, combining them is expected to give the best performance. A simple way to incorporate external factors into recommender systems is through a linear model, which means to augment the latent factors from collaborative filtering method with the extracted social factors (as shown in Fig.2). The augmented user factor will be treated as the representations of social network and rating preference. Then, the score function of predicting the preference of user u to item i can be modified as:

$$\widehat{r}_{u,i}(\boldsymbol{U}, \boldsymbol{I}) = \boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u,$$

where $\boldsymbol{W}_u \in \mathbb{R}^d$ is the weight vector that transforms the learned social network representations from the neural networks into the collaborative filtering latent space of user u. Its value indicates how much the pretrained network features should contribute to user u. To avoid overfitting, the corresponding regularization term is added into (1). Hence, the objective function of our social embedding based recommendation method MFn2v can be arrived as:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}) \\
&= \min_{\boldsymbol{U},\boldsymbol{I},\boldsymbol{W}} \frac{1}{2} \sum_{u=1}^{M} \sum_{i=1}^{N} (r_{u,i} - \boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i - \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u)^2 + \\
&\quad \frac{\lambda_{\boldsymbol{U}}}{2} \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{I}}}{2} \|\boldsymbol{I}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{W}}}{2} \|\boldsymbol{W}\|_{\mathrm{F}}^2, \qquad (2)
\end{aligned}$$

where $\lambda_{\boldsymbol{W}}$ is the regularization parameter of $\boldsymbol{W}$.

We apply stochastic gradient descent method to find a local minimum of (2), and update the latent factors $\boldsymbol{U}$, $\boldsymbol{I}$ and $\boldsymbol{W}$ by the following gradients:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W})}{\partial \boldsymbol{U}_u} \\
&= \sum_{i=1}^{N} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u - r_{u,i}) \boldsymbol{I}_i + \lambda_{\boldsymbol{U}} \boldsymbol{U}_u, \\
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W})}{\partial \boldsymbol{I}_i} \\
&= \sum_{u=1}^{M} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u - r_{u,i}) \boldsymbol{U}_u + \lambda_{\boldsymbol{I}} \boldsymbol{I}_i, \\
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W})}{\partial \boldsymbol{W}_u} \\
&= \sum_{i=1}^{N} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u - r_{u,i}) \boldsymbol{X}_u + \lambda_{\boldsymbol{W}} \boldsymbol{W}_u.
\end{aligned}$$

②The parameter settings of our pre-trained node2vec model are: $d = 10$; $l = 80$; $r = 10$; $k = 10$; $p = 1$; $q = 0.5$.

4.3 MFn2v+ Model

MFn2v improves traditional recommender systems by assuming that users with social connections will have similar preferences, but it only considers the recommendation problem from a user's perspective and the item sequential relationship is ignored (it assumes items are i.i.d.), which is unreasonable in many scenarios[62]. For example, in a product recommendation scenario, when a user bought a football in the past, he/she will probably buy the soccer shoes later. Items that are bought in a short time sequence have a strong correlation with each other. Based on this intuition, we construct the item correlation according to the following data policy. If two items are rated by the same user in a short time interval③ (see ΔT), then we assume that there is a correlation between the two items. For the items that are rated in a long time interval or not rated by the same user, we cannot infer any correlations. Hence, the definition of item sequential relation network is shown as follows.

Definition 2 (Item Relation Network). *The item relation network is denoted by $\mathcal{G}_{ii} = (\mathcal{I}, \mathcal{E}_{ii})$, where $\mathcal{I}$ represents the item set, and $\mathcal{E}_{ii}$ represents the edge set. Given a time interval ΔT, for any item pair $\{(i_q, t_q), (i_n, t_n)\}$ rated by the same user ($i_q \in \mathcal{I}, i_n \in \mathcal{I}, t_q$ and t_n are the time of being rated), if $0 < t_n - t_q \leqslant \Delta T$, there will be an edge $e_{qn} \in \mathcal{E}_{ii}$ from i_q to i_n; otherwise none.*

Item relation network captures the sequential characteristics of users' rating behaviours. Intuitively, if two products i_q and i_n have similar sequential patterns, when we know that a user has bought i_q, we can predict that the user will probably buy i_n in future, as it has a strong correlation with i_q (such as football and soccer shoes). To capture the item sequential patterns, we pre-train node2vec④ to learn the sequence representations of items from the item relation network, and let $\boldsymbol{Y}_i \in \mathbb{R}^d$ be the extracted factors of item i. To consider the item embeddings for recommendation, similar to MFn2v, we further incorporate this external representation by a linear model. Then, the rating function $\widehat{r}_{u,i}(\boldsymbol{U}, \boldsymbol{I})$ can be rewritten as:

$$\widehat{r}_{u,i}(\boldsymbol{U}, \boldsymbol{I}) = \boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u + \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i,$$

where $\boldsymbol{V}_i \in \mathbb{R}^d$ is the weighted vector that indicates how much the pre-trained network feature should contribute to item i. To avoid overfitting, the regularization term of $\boldsymbol{V}$ is also added, and the objective function of our final embedding-based recommendation method MFn2v+ is achieved:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V}) \\
&= \min_{\boldsymbol{U},\boldsymbol{I},\boldsymbol{W},\boldsymbol{V}} \frac{1}{2} \sum_{u=1}^{M} \sum_{i=1}^{N} (r_{u,i} - \boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i - \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u - \\
&\quad \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i)^2 + \frac{\lambda_{\boldsymbol{U}}}{2} \|\boldsymbol{U}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{I}}}{2} \|\boldsymbol{I}\|_{\mathrm{F}}^2 + \\
&\quad \frac{\lambda_{\boldsymbol{W}}}{2} \|\boldsymbol{W}\|_{\mathrm{F}}^2 + \frac{\lambda_{\boldsymbol{V}}}{2} \|\boldsymbol{V}\|_{\mathrm{F}}^2, \qquad (3)
\end{aligned}$$

where $\lambda_{\boldsymbol{U}}, \lambda_{\boldsymbol{I}}, \lambda_{\boldsymbol{W}}$ and $\lambda_{\boldsymbol{V}}$ are the regularization parameters. As optimized for MFn2v, the local minimum of this equation can be found by the stochastic gradient descent method. The gradients of the latent factors $\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ are:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V})}{\partial \boldsymbol{U}_u} \\
&= \lambda_{\boldsymbol{U}} \boldsymbol{U}_u + \sum_{i=1}^{N} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u + \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i - r_{u,i}) \boldsymbol{I}_i, \qquad (4)
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V})}{\partial \boldsymbol{I}_i} \\
&= \lambda_{\boldsymbol{I}} \boldsymbol{I}_i + \sum_{u=1}^{M} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u + \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i - r_{u,i}) \boldsymbol{U}_u, \qquad (5)
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V})}{\partial \boldsymbol{W}_u} \\
&= \lambda_{\boldsymbol{W}} \boldsymbol{W}_u + \sum_{i=1}^{N} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u + \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i - r_{u,i}) \boldsymbol{X}_u, \qquad (6)
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{U}, \boldsymbol{I}, \boldsymbol{W}, \boldsymbol{V})}{\partial \boldsymbol{V}_i} \\
&= \lambda_{\boldsymbol{V}} \boldsymbol{V}_i + \sum_{u=1}^{M} (\boldsymbol{U}_u^{\mathrm{T}} \boldsymbol{I}_i + \boldsymbol{W}_u^{\mathrm{T}} \boldsymbol{X}_u + \boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{Y}_i - r_{u,i}) \boldsymbol{Y}_i. \qquad (7)
\end{aligned}$$

Algorithm 1 presents the training process of our MFn2v+ method.

5 Experimental Results and Analysis

In this section, we conduct several experiments on two real-world datasets to evaluate the performance of our proposed method. To make the evaluation results

③In our work, ΔT is set to 24 hours.

④The parameter settings of node2vec are: $d = 10$; $l = 80$; $r = 10$; $k = 10$; $p = 1$; $q = 0.5$.

more clearly, we also conduct experiments to evaluate our embedding-based method with different parameter settings.

Algorithm 1. Learn($\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{W}$, $\boldsymbol{V}$)

1: **Input:** user rating matrix $\boldsymbol{R}$, learning rate η, regularization parameters $\lambda_{\boldsymbol{U}}$, $\lambda_{\boldsymbol{I}}$, $\lambda_{\boldsymbol{W}}$, $\lambda_{\boldsymbol{V}}$, the extracted network factors $\boldsymbol{X}$ and $\boldsymbol{Y}$
2: **Output:** latent factors $\boldsymbol{U}$, $\boldsymbol{I}$, and weight matrixes $\boldsymbol{W}$, $\boldsymbol{V}$
3: Initialize $\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{W}$ and $\boldsymbol{V}$
4: **do**
5: Randomly select one training example (u, i) from $\boldsymbol{R}$
6: Update $\boldsymbol{U}_u$ according to (4)
7: Update $\boldsymbol{I}_i$ according to (5)
8: Update $\boldsymbol{W}_u$ according to (6)
9: Update $\boldsymbol{V}_i$ according to (7)
10: Calculate $\mathcal{L}(t)$ (the loss error in t step) by utilizing (3)
11: **while** $\mathcal{L}(t)-\mathcal{L}(t-1) >$ tolerate error
12: **return** $\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{W}$ and $\boldsymbol{V}$

5.1 Datasets

In experiments, we utilize two popular social networks⑤ Ciao and Epinions to evaluate our proposed method. The Ciao dataset is crawled from the online review site Ciao.com⑥, which is a multi-million-strong online community that provides a forum for registered users to write reviews and give their opinions on a wide variety of products to help others make decisions. The Epinions dataset comes from another well-known product review website Epinions.com⑦, which was launched in 1999. At Epinions, visitors are also allowed to read new and old reviews about a variety of items from other users to help them decide on a purchase. On both Ciao and Epinions, registered users express their opinions by rating the product or service using a 5-scale integer (from 1 to 5), and maintain a trust list to determine in which order the product views are shown to visitors. Users will establish trust relation with the users whose reviews are interesting and helpful for them.

The datasets we use are public published by Tang *et al.*[34], including the user-product rating data, the rating timestamp, and the social trust network among users. As the rating data is very sparse, the recommendation problem in these two datasets is challenging. The work we study in this paper is how to improve the traditional recommendation methods by utilizing the social network and the item sequential patterns. The statistics of these two datasets are summarized in Table 1.

Table 1. Statistics of Ciao and Epinions

Statistics	Ciao	Epinions
Number of users	7 375	22 166
Number of items	106 797	296 277
Number of ratings	284 086	922 267
Rating density	0.000 4	0.000 2
Number of trust relations	111 781	355 813
Trust relation density	0.004 1	0.001 4

5.2 Evaluation Metrics

In this work, we use the mean absolute error (MAE) and the root mean square error (RMSE)[26] to evaluate the recommendation performance of our proposed approach. The metric MAE is defined as:

$$MAE = \frac{1}{N}\sum_{u,i}|r_{u,i} - \widehat{r}_{u,i}|,$$

where $r_{u,i}$ is the real rating value that user u rated to item i, and $\widehat{r}_{u,i}$ is the corresponding predicted rating value. N denotes the rating number used for test. The evaluation metric RMSE we use in this work is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(r_{u,i} - \widehat{r}_{u,i})^2}.$$

As MAE and RMSE measure the prediction error of the recommendation method, a lower value of MAE and RMSE indicates the method can predict more accurately.

5.3 Comparison Results

In order to evaluate the recommendation performance of our proposed method, we conduct several comparisons with the following related methods.

MF. This is a basic recommendation method[1], which seeks to approximate the rating matrix by a multiplication of two low-rank factors. But only rating data is used in this method.

LFL. This is the log-linear model with latent features proposed for dyadic prediction[20], where the side information can be easily exploited for predictions.

SocialMF. This is a state-of-the-art social recommendation method[32], which considers the influence

⑤http://www.cse.msu.edu/tangjili//trust.html, May 2018.
⑥http://www.ciao.com, May 2018.
⑦http://www.epinions.com, May 2018.

of direct neighbors to the users and incorporates the mechanism of trust propagation for better recommendations.

SoReg. This is another popular recommendation method that models the social network information as social regularization terms to constrain the matrix factorization objective function[33].

In experiments, we evaluate these above algorithms by splitting the training data with different ratios (from 50% to 90%), where training data 90% indicates that we randomly select 90% user-item ratings of each user for training, and utilize the remaining 10% rating data for prediction. To ensure the validity of this data, we conduct the random selection five times independently. For Ciao, the regularization parameters of $\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{W}$ and $\boldsymbol{V}$ are set as $\lambda_{\boldsymbol{U}} = \lambda_{\boldsymbol{I}} = 0.6, \lambda_{\boldsymbol{W}} = \lambda_{\boldsymbol{V}} = 0.001$. For Epinions, the values of $\lambda_{\boldsymbol{U}}, \lambda_{\boldsymbol{I}}, \lambda_{\boldsymbol{W}}$, and $\lambda_{\boldsymbol{V}}$ are set as $\lambda_{\boldsymbol{U}} = \lambda_{\boldsymbol{I}} = 0.6, \lambda_{\boldsymbol{W}} = 0.001, \lambda_{\boldsymbol{V}} = 0.005$. For both Ciao and Epinions, the dimension of the latent factors $\boldsymbol{U}$ and $\boldsymbol{I}$ is set as $k = 15$. The dimension of the weight matrixes $\boldsymbol{W}$ and $\boldsymbol{V}$ for extracted features is set as $d = 10$.

The comparison results are shown in Table 2 (80% rating data for training and remaining 20% for prediction), where Δ denotes the improvement of our MFn2v+ model over other corresponding approaches. From this result, we can observe that as MF only uses the rating information for recommendations, it does worse than the other related methods on both Ciao and Epinions data. The LFL method that can incorporate side information for recommendations can work better than MF. SocialMF as a state-of-the-art social recommendation method achieves a slightly better performance than another popular recommendation SoReg. Both of these two social recommendation methods can perform better than MF and LFL, which demonstrates the existence of the social influence, and incorporating it is helpful for us to make more accurate prediction. From this result, we can also find that our proposed method MFn2v performs better than SocialMF, which indicates that the pre-trained network embeddings are good representations of users and fusing them with latent factors can effectively model both the users' personal and social interest. In this result, MFn2v+ outperforms MFn2v and reaches the best performance on both two datasets, which denotes that the item sequential information is important in this social recommendation scenario, and considering the extracted item embeddings from sequential relation network can help to improve the recommendation performance. From Table 2, we also notice that as the Epinions data is much sparser, all the methods on Ciao can reach a better performance than those on Epinions.

Table 2. Comparison Results on Ciao and Epinions (Training = 80%, $k = 15$, $d = 10$)

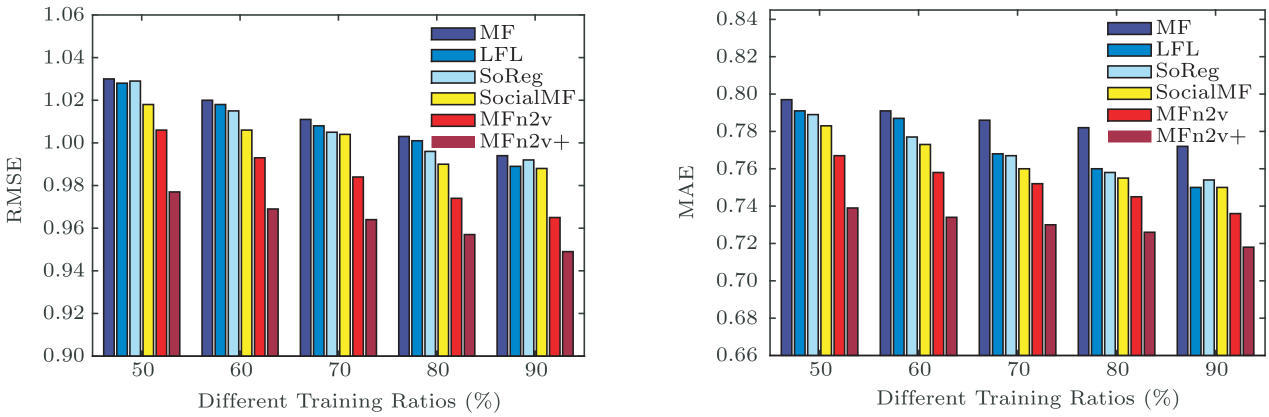
Method	Ciao		Epinions	
	MAE/Δ(%)	RMSE/Δ(%)	MAE/Δ(%)	RMSE/Δ(%)
MF	0.782/7.29	1.003/4.59	0.846/5.44	1.086/4.14
LFL	0.760/4.61	1.001/4.40	0.842/4.99	1.070/2.71
SoReg	0.758/4.35	0.996/3.91	0.839/4.60	1.066/2.30
SocialMF	0.755/3.97	0.990/3.33	0.830/3.61	1.062/1.98
MFn2v	0.745/2.68	0.974/1.75	0.824/2.91	1.060/1.79
MFn2v+	0.725/-	0.957/-	0.800/-	1.041/-

5.4 Impact of Different Training Data Settings

To investigate the performance of our proposed method MFn2v+ on cold-start situations, we evaluate our method with different training data settings (from 50% to 90%). For example, 50% indicates only 50% of the training data is selected for training. The item sequential information and the social network information are available for all the training settings. The experimental results evaluated by MAE and RMSE are shown in Fig.3 and Fig.4, from which we can observe that MFn2v and MFn2v+ are generally superior to all the baseline methods in our experiments. The MFn2v+ method that incorporates both social connections and item sequential patterns achieves the best performance in all different training ratios. This result demonstrates the importance of the extracted embeddings for the absence of usage data, and our method can be suitable for cold-start settings. From Fig.3 and Fig.4, we also notice that, as the rating data used for training decreases, the improvement of our MFn2v+ method becomes more obvious, which indicates that exploiting the pre-trained network representations for predictions is more effective.

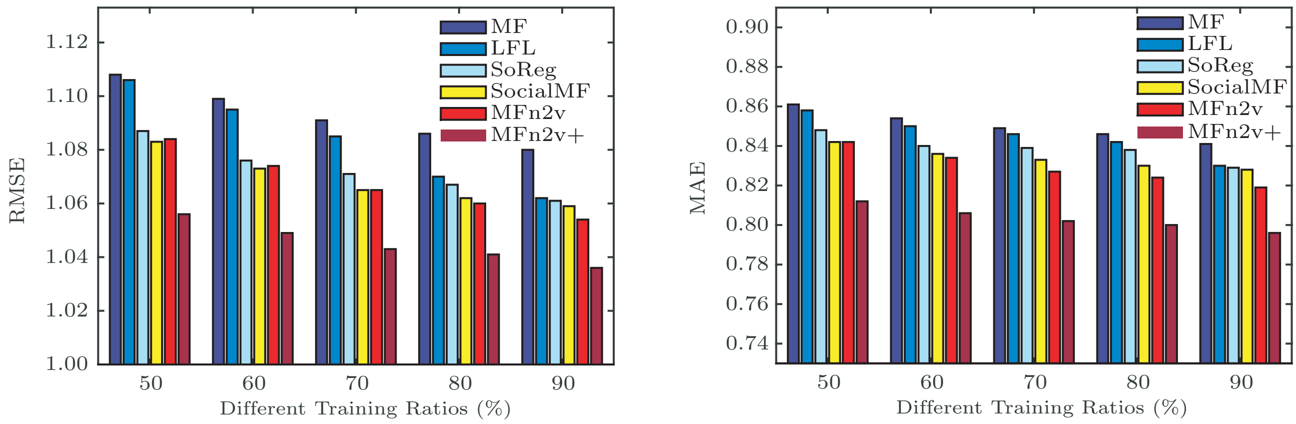
5.5 Impact of Latent Factor Dimension k

The dimension of the latent factors controls how much information can be utilized to represent users and items from the collaborative filtering model. If the dimension is too small, the learned factors will not be enough to represent the user interest. If the dimension is too large, not only will the feature appear to be duplicated, but also the model is more likely to over-fit (as the model is too complicated). Appropriate factor



(a) (b)

Fig.3. Impact of different training ratios on Ciao.



(a) (b)

Fig.4. Impact of different training ratios on Epinions.

dimension plays a crucial role for us to make the right predictions.

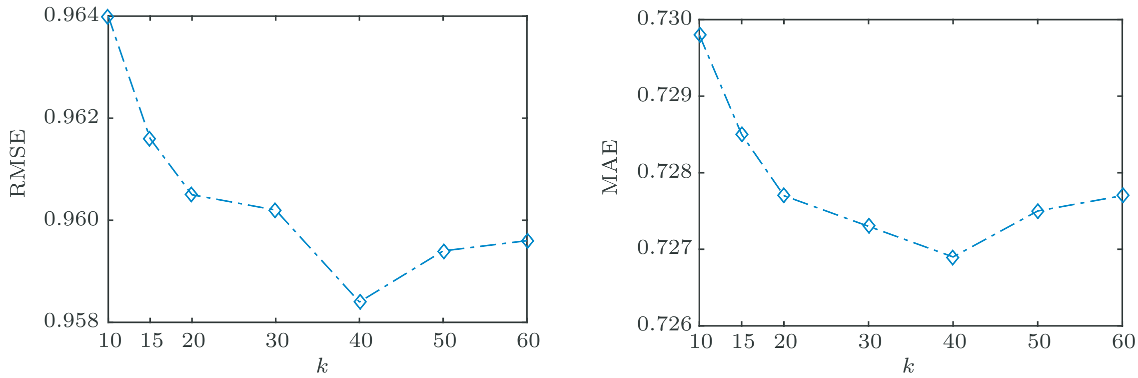
Fig.5 and Fig.6 show the experimental results with different dimension values. From the results, we can observe that as the factor dimension increases, the test error decreases at first, but when k surpasses a certain value (for Ciao is 40, for Epinions is 70), the test error increases with the further increase of the value of k, which demonstrates the intuition that we have mentioned above. In experiments, we also find that with relative small factor dimensions, MFn2v+ can achieve a reasonable result. Hence, to reduce the computational complexity, we set the latent factor dimension k to 15 on both two datasets.

5.6 Convergence Analysis

To explore the efficiency of our embedding-based method, we further conduct experiments to compare the convergence of our MFn2v+ method with MF on Ciao and Epinions. To make them comparable, the same learning rates are adopted (80 on Ciao, and 300 on Epinions). The comparison results are shown in Fig.7 and Fig.8, from which we can observe that both of these two methods converge very fast (they converge within 80 and 50 iterations on Ciao and Epinions, respectively). Compared with MF, the convergence rate of MFn2v+ is not slowed down by incorporating the embeddings from social connections and item sequential patterns, and on the contrary it can make a better performance than the MF method on both Ciao and Epinions.

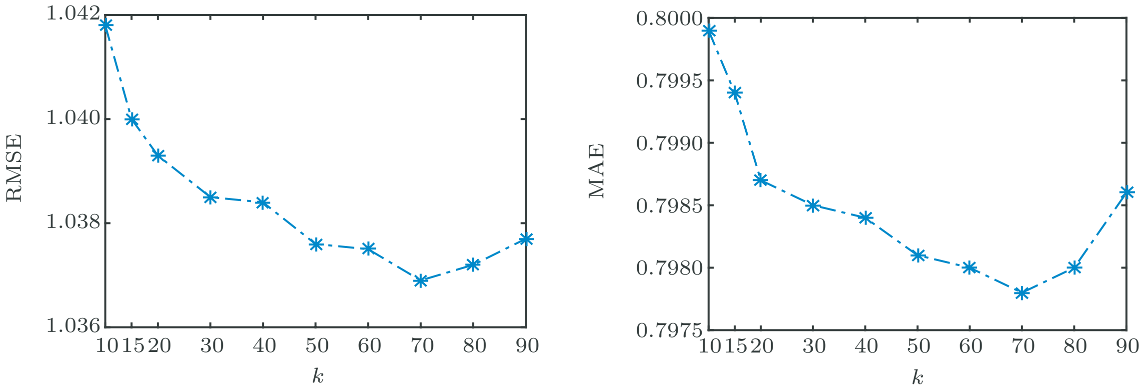
6 Conclusions

To better utilize the latent information hidden in the users' social network and rating patterns for recommendations, this work proposed a two-step embedding-based recommendation method, which can exploit the



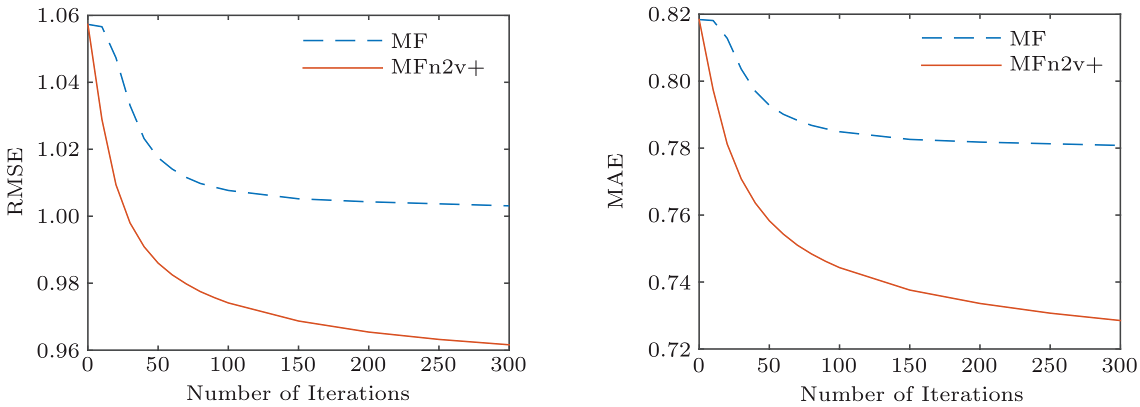
(a) (b)

Fig.5. Impact of latent factor dimension k on Ciao.



(a) (b)

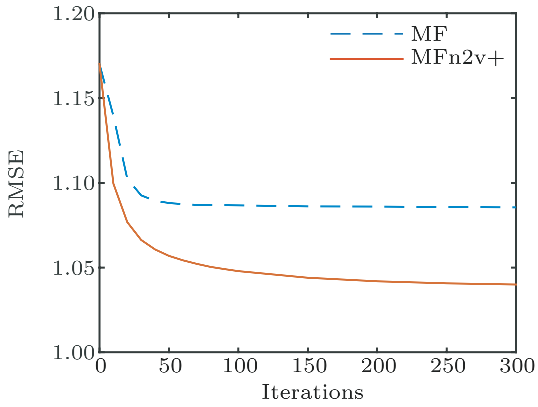
Fig.6. Impact of latent factor dimension k on Epinions.



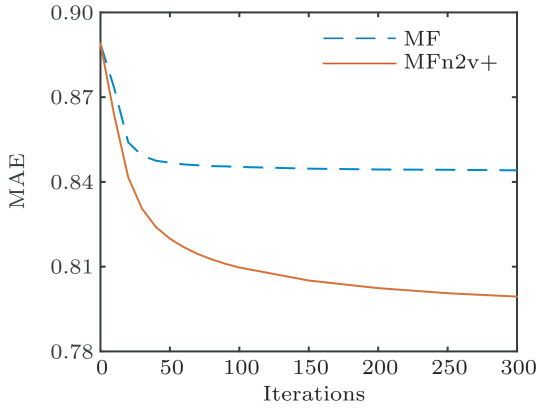
(a) (b)

Fig.7. Convergence analysis on the Ciao data.

advantages of the network embedding model and the collaborative filtering model simultaneously. Specifically, we first pre-trained a neural network based embedding model to learn the high-level representations of users and items, and then we fused these learned features with the latent factors from the collaborative filtering model linearly. Compared with existing work, our proposed method can take advantage of the neural network based embedding model and can exploit the deep structure of the network information rather than



(a)



(b)

Fig.8. Convergence analysis on the Epinions data.

just regard them as regularization terms. Experimental results on two real-world datasets demonstrated the importance of these pre-learned factors, and the effectiveness of our proposed methods.

This work mainly investigated the effectiveness of extracted social and item representations in traditional recommendation task (rating prediction), and their influence on more complex tasks (such as long tail recommendation and session-based recommendation) has not been explored. In future we plan to leverage these pre-trained network embeddings to make predictions in more complex recommendation problems.

References

[1] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8): 30-37.

[2] Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In *Proc. the 20th International Conference on Neural Information Processing Systems*, December 2007, pp.1257-1264.

[3] Ma H, Yang H, Lyu M R, King I. SoRec: Social recommendation using probabilistic matrix factorization. In *Proc. the 17th ACM Conference on Information and Knowledge Management*, October 2008, pp.931-940.

[4] Ma H, King I, Lyu M R. Learning to recommend with social trust ensemble. In *Proc. the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2009, pp.203-210.

[5] Ma H, Zhou D, Liu C, Lyu M R, King I. Recommender systems with social regularization. In *Proc. the 4th ACM International Conference on Web Search and Data Mining*, February 2011, pp.287-296.

[6] Guo L, Ma J, Jiang H R, Chen Z M, Xing C M. Social trust aware item recommendation for implicit feedback. *Journal of Computer Science and Technology*, 2015, 30(5): 1039-1053.

[7] Wang W, Yin H, Sadiq S, Chen L, Xie M, Zhou X. SPORE: A sequential personalized spatial item recommender system. In *Proc. the 32nd IEEE International Conference on Data Engineering*, May 2016, pp.954-965.

[8] Chu W T, Tsai Y L. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web*, 2017, 20(6): 1313-1331.

[9] Ling D, Zhan M, Ellis D P W. Content-aware collaborative music recommendation using pre-trained neural networks. In *Proc. the International Symposium/Conference on Music Information Retrieval*, Oct. 2015, pp.295-301.

[10] Zhao W X, Huang J, Wen J R. Learning distributed representations for recommender systems with a network embedding approach. In *Proc. the Asia Information Retrieval Symposium*, October 2016, pp.224-236.

[11] Perozzi B, Alrfou R, Skiena S. DeepWalk: Online learning of social representations. In *Proc. the Asia Information Retrieval Symposium*, August 2014, pp.701-710.

[12] Grover A, Leskovec J. Node2Vec: Scalable feature learning for networks. In *Proc. the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2016, pp.855-864.

[13] Goldberg D, Nichols D, Oki B M, Terry D. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 1992, 35(12): 61-70.

[14] Ma H, King I, Lyu M R. Effective missing data prediction for collaborative filtering. In *Proc. the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2007, pp.39-46.

[15] Li D, Lv Q, Shang L, Gu N. Item-based top-N recommendation resilient to aggregated information revelation. *Knowledge-Based Systems*, 2014, 67(3): 290-304.

[16] Canny J. Collaborative filtering with privacy via factor analysis. In *Proc. the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, August 2002, pp.238-245.

[17] Kamishima T, Akaho S, Asoh H, Sato I. Model-based approaches for independence-enhanced recommendation. In *Proc. the 16th IEEE International Conference on Data Mining Workshops*, December 2017, pp.860-867.

[18] Yin H, Zhou X, Cui B, Wang H, Zheng K, Nguyen Q V H. Adapting to user interest drift for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(15): 2566-2581.

[19] Hofmann T. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *Proc. the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, July 2003, pp.259-266.

[20] Menon A K, Elkan C. A log-linear model with latent features for dyadic prediction. In *Proc. the IEEE International Conference on Data Mining*, December 2010, pp.364-373.

[21] Massa, P, Avesani P. Trust-aware recommender systems. In *Proc. the ACM Conference on Recommender Systems*, October 2007, pp.17-24.

[22] Jamali M, Ester M. TrustWalker: A random walk model for combining trust-based and item-based recommendation. In *Proc. the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, June 2009, pp.397-406.

[23] Cao G, Kuang L. Identifying core users based on trust relationships and interest similarity in recommender system. In *Proc. the 2016 IEEE International Conference on Web Services*, June 2016, pp.284-291.

[24] Li W, Ye Z, Xin M, Jin Q. Social recommendation based on trust and influence in SNS environments. *Multimedia Tools and Applications*, 2017, 76(9): 11585-11602.

[25] Yin H, Hu Z, Zhou X, Wang H, Zheng K, Nguyen Q V H, Sadiq S. Discovering interpretable geo-social communities for user behavior prediction. In *Proc. IEEE International Conference on Data Engineering*, May 2016, pp.942-953.

[26] Ma H, King I, Lyu M R. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology*, 2011, 2(3): 1-19.

[27] Kannan R, Ishteva M, Park H. Bounded matrix factorization for recommender system. *Knowledge and Information Systems*, 2014, 39(3): 491-511.

[28] Yu H, Hsieh C, Si S, Dhillon I S. Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, 2014, 41(3): 793-819.

[29] Zhao Z L, Wang C D, Wan Y Y, Lai J H, Huang D. FTMF: Recommendation in social network with feature transfer and probabilistic matrix factorization. In *Proc. the International Joint Conference on Neural Networks*, July 2016, pp.847-854.

[30] Liu Y, Wei W, Sun A, Miao C. Exploiting geographical neighborhood characteristics for location recommendation. In *Proc. the 23rd ACM International Conference on Information and Knowledge Management*, November 2014, pp.739-748.

[31] Ma H. An experimental study on implicit social recommendation. In *Proc. the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2013, pp.73-82.

[32] Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. the 4th ACM Conference on Recommender Systems*, September 2010, pp.135-142.

[33] Ma H, Zhou D, Liu C, Lyu M R, King I. Recommender systems with social regularization. In *Proc. the 4th ACM International Conference on Web Search and Data Mining*, February 2011, pp.287-296.

[34] Tang J, Hu X, Gao H, Liu H. Exploiting local and global social context for recommendation. In *Proc. the 23rd International Joint Conference on Artificial Intelligence*, August 2013, pp.2712-2718.

[35] Koren Y. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 2010, 53(4): 89-97.

[36] Xiong L, Chen X, Huang T K, Schneider J G, Carbonell J G. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *Proc. the SIAM International Conference on Data Mining*, April 2010, pp.211-222.

[37] Liu N N, He L, Zhao M. Social temporal collaborative ranking for context aware movie recommendation. *ACM Transactions on Intelligent Systems and Technology*, 2013, 4(1): 1-26.

[38] Lian D, Zhang Z, Ge Y, Zhang F, Yuan N J, Xie X. Regularized content-aware tensor factorization meets temporal-aware location recommendation. In *Proc. IEEE International Conference on Data Mining*, December 2016, pp.1029-1034.

[39] Yin H, Cui B. Spatio-Temporal Recommendation in Social Media. Springer Singapore, 2016.

[40] Zhang J D, Chow C Y. TICRec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations. *IEEE Transactions on Services Computing*, 2016, 9(4): 633-646.

[41] Zhang J D, Chow C Y, Li Y. LORE: Exploiting sequential influence for location recommendations. In *Proc. the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, November 2014, pp.103-112.

[42] Yin H, Cui B, Zhou X, Wang W, Huang Z, Sadiq S. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *ACM Transactions on Information Systems*, 2016, 35(2): Atricle No. 11.

[43] Yin H, Chen H, Sun X, Wang H, Wang Y, Nguyen Q V H. SPTF: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In *Proc. IEEE International Conference on Data Mining*, November 2017, pp.585-594.

[44] Tong Y, Chen L, Zhou Z, Jagadish H V, Shou L, Lv W. SLADE: A smart large-scale task decomposer in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering*, 2018. (to be appeared)

[45] Tong Y, She J, Ding B, Wang L, Chen L. Online mobile micro-task allocation in spatial crowdsourcing. In *Proc. IEEE International Conference on Data Engineering*, May 2016, pp.49-60.

[46] Angela C R, Marl V O, Gensel J, Martin H. Contextual user profile for adapting information in nomadic environments. In *Proc. the International Conference on Web Information Systems Engineering*, December 2007, pp.337-349.

[47] Lian D, Zhao C, Xie X, Sun G, Chen E, Rui Y. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proc. the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2014, pp.831-840.

[48] Zheng Y, Burke R, Mobasher B. The role of emotions in context-aware recommendation. In *Proc. the 3rd Workshop on Human Decision Making in Recommender Systems*, October 2013, pp.21-28.

[49] Lecun Y, Bengio Y, Hinton G. Deep learning. *Proceedings of the VLDB Endowment*, 2015, 521(7553): 436-444.

[50] Nguyen T T, Lauw H W. Representation learning for homophilic preferences. In *Proc. the 10th ACM Conference on Recommender Systems*, September 2016, pp.317-324.

[51] Deng S, Huang L, Xu G, Wu X, Wu Z. On deep learning for trust-aware recommendations in social networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 28(5): 1164-1177.

[52] Yin H, Wang W, Wang H, Chen L, Zhou X. Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(11): 2537-2551.

[53] Kataria S, Agarwal A. Distributed representations for content-based and personalized tag recommendation. In *Proc. the IEEE International Conference on Data Mining Workshop*, November 2015, pp.1388-1395.

[54] Yang C, Sun M, Zhao W Xin, Liu Z, Chang E Y. A neural network approach to jointly modeling social networks and mobile trajectories. *ACM Transactions on Information Systems*, 2016, 35(4): 36:1-36:28.

[55] Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. Learning graph-based POI embedding for location-based recommendation. In *Proc. the 25th ACM International Conference on Information and Knowledge Management*, October 2016, pp.15-24.

[56] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In *Proc. the 26th International Conference on Neural Information Processing Systems*, October 2013, pp.3111-3119.

[57] Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. LINE: Large-scale information network embedding. In *Proc. the 24th International Conference on World Wide Web*, May 2015, pp.1067-1077.

[58] Liang D, Altosaar J, Charlin L, Blei D M. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proc. the 10th ACM Conference on Recommender Systems*, September 2016, pp.59-66.

[59] Zhao W X, Li S, He Y, Chang E Y, Wen J R, Li X. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(5): 1147-1159.

[60] Rennie J D M, Srebro N. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. the 22nd International Conference on Machine Learning*, August 2005, pp.713-719.

[61] Menon A K, Elkan C. Link prediction via matrix factorization. In *Proc. the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, September 2011, pp.437-452.

[62] Guo L, Ma J, Chen Z, Jiang H. Learning to recommend with social relation ensemble. In *Proc. the 21st ACM International Conference on Information and Knowledge Management*, October 2012, pp.2599-2602.

Lei Guo received his Ph.D. degree in computer architecture from Shandong University, Jinan, in 2015. Now he is a lecturer and master supervisor in the School of Management Science and Engineering, Shandong Normal University, Jinan. His research interests include information retrieval, social network, and recommender system.

Yu-Fei Wen is a Master candidate in the School of Management Science and Engineering, Shandong Normal University, Jinan. Her research interests include deep neural network and recommender system.

Xin-Hua Wang received his M.S. degree in applied mathematics from Dalian University of Technology, Dalian, in 2001, and his Ph.D. degree in management science and engineering from Shandong Normal University, Jinan, in 2008. Currently, he is a professor and master supervisor in the School of Information Science and Engineering, Shandong Normal University, Jinan. His research interests include distributed network and recommender system.