Computing and Informatics, Vol. 44, 2025, 223–244, doi: 10.31577/cai_2025_1_223

LOCAL MATRIX FACTORIZATION WITH NETWORK EMBEDDING FOR RECOMMENDER SYSTEMS

Jinmao Xu

Henan University of Engineering
Xianghe Road 1
451191 Zhengzhou, China
e-mail: xujinmao1987@163.com

Zhifeng Liu

Zhongyuan University of Technology
Zhongyuan Road 41
450007 Zhengzhou, China
e-mail: nineteen109@outlook.com

Lei Tan

Henan Key Laboratory of Cyberspace Situation Awareness
Road Kexue 100
Zhengzhou, China
e-mail: tanlei08@sina.com

Tianrui Li*, Tianqiang Peng

Henan University of Engineering
Xianghe Road 1
451191 Zhengzhou, China
e-mail: tianruili827@163.com, 447805213@qq.com

Daofu Gong

Henan Key Laboratory of Cyberspace Situation Awareness
Road Kexue 100
Zhengzhou, China
e-mail: gongdf@aliyun.com

Abstract. In recommender systems, the rating matrix is usually not a global low-rank but local low-rank. Constructing low-rank submatrices for matrix factorization can improve the accuracy of rating prediction. This paper proposes a novel network embedding-based local matrix factorization model, which can built more meaningful sub-matrices. To alleviate the sparsity of the rating matrix, the social data and the rating data are integrated into a heterogeneous information network, which contains multiple types of objects and relations. The network embedding algorithm extracts the node representations of users and items from the heterogeneous information network. According to the correlation of the node representations, the rating matrix is divided into different sub-matrices. Finally, the matrix factorization is performed on the sub-matrices for rating prediction. We test our network embedding-based method on two real-world public data sets (Yelp and Douban). Experimental results show that our method can obtain more accurate prediction ratings.

Keywords: Matrix factorization, network embedding, local low-rank, recommender systems

Mathematics Subject Classification 2010: 68-T99

1 INTRODUCTION

Recommender systems (RS) could deal with the problem of information overload [1] in the big data era, which has been widely studied [2, 3, 4]. By analyzing the previous user-item interactions (e.g. rating data and browsing data), recommender systems can learn the preferences of users, which is utilized to predict the user behaviors for personalized recommendations [5]. Generally, to facilitate the representation of user preferences and item attributes, users and items are mapped into a low-dimensional vector space. However, it is challenging to develop an effective approach to characterize users and items in recommender systems [6].

As one of the most widely used collaborative filtering methods, matrix factorization [7, 8] has received much attention for its good performance and scalability. By factorizing the user-item interaction matrix into two matrices, namely the user latent factor matrix and the item latent factor matrix, matrix factorization maps users and items into a latent factor space. As a result, user preferences and item attributes are associated with latent factor vectors. However, the basic assumption of the matrix factorization requires that the interaction matrix is low-rank, which means that the users (items) in the matrix are highly correlated [9]. The rating matrix is usually not a global low-rank matrix [10], which means not all users have similar preferences. Since the matrix is locally stable [9], matrix factorization can be

* Corresponding author

performed in a local low-rank matrix to achieve higher prediction accuracy. Generally, according to user preferences (item attributes), all users (items) can be divided into different subsets. In the subgroup, there is a closer correlation between the users (items). Therefore, the user-item interaction matrices constructed by the user subsets and item subsets are low-rank. The whole rating matrix can be converted to multiple local low-rank matrices. It is worth noting that global low-rank refers to the properties of the entire rating matrix, and local low-rank refers to the properties of the sub matrix. Recommending in the local low-rank matrix can get better performance. To effectively select anchor points for sub-matrices construction, Zhang et al. propose a heuristic method to select anchor points [11]. According to the social homophily theory, Zhao et al. exploit users' social connections to construct meaningful sub-matrices [12].

Most users within commercial platforms rarely rate [13]. Therefore, user latent factor vectors extracted by matrix factorization cannot effectively reflect user preferences, which will degrade the recommendation performance. For example, Lee et al. propose a local low-rank matrix factorization method, which factorizes the rating matrix to get the latent factor vectors of users and items. According to the correlation of these latent factor vectors, the users and the items are constructed as a sub-matrix [10]. However, the latent factor vector of users who rate fewer items cannot accurately characterize the preferences of these users. Hence, the sub-matrices established by these latent factor vectors are meaningless.

Based on the above intuition, we propose a local matrix factorization model based on network embedding, called LMFE. To accurately characterize the users and the items, auxiliary data is added to the model. Auxiliary data generally includes user attributes, item attributes, social relations, and other information, which characterize users and items from multiple aspects [14]. Previous studies [15] have proved that the auxiliary data can effectively improve recommendation performance.

How to use auxiliary data to improve the accuracy of user preference prediction is a challenge for recommendation systems. The auxiliary data either contains multiple types of objects or multiple types of relations. Based on these characteristics of the auxiliary data, we utilize the Heterogeneous Information Network (HIN) [15, 16] to model auxiliary data. Hence, the auxiliary data and the rating data are modeled as a heterogeneous information network in this paper. Finally, the user representations and item representations are obtained by applying a novel embedding method to the HIN.

Since the rating matrix is local low-rank, it can be divided into multiple low-rank sub-matrices. Moreover, the number of sub-matrices is determined by the number of selected anchor points (user-item pairs) namely an anchor point corresponds to a sub-matrix. By calculating the correlation between anchor points and all data points according to the user representations and item representations, data points are classified into sub-matrices. Since the sub-matrices constructed by the data points with high correlation are low-rank, matrix factorization on these sub-matrices can improve the recommendation performance. LMFE applies to group recommendation scenarios. That is, multiple groups exist within many users, and users in these groups have similar preferences. Therefore, we use auxiliary information and ratings to divide user groups and make recommendations in the subspace, improving the accuracy of rating prediction.

The contributions of this paper are summarized as follows:

1. We present a novel HIN embedding method to learn the node representations, which can accurately characterize the user preferences and item attributes.
2. We propose a network embedding based local matrix factorization model, which can construct low-rank sub-matrices effectively, and improve the performance of recommender systems.
3. We conduct experiments on two real-world datasets and demonstrate the effectiveness of LMFE.

2 PRELIMINARY

In this section, we define the notations used in this article and introduce some preliminary knowledge.

2.1 Heterogeneous Information Networks

Definition 1. Heterogeneous Information Networks (HIN) [16]. A HIN is defined as a graph $G = (\nu, \varepsilon)$ with an object type mapping function $\tau : \nu \to A$ and a link type mapping function $\phi : \varepsilon \to R$, where each object $v \in V$ belongs to one particular object type $\tau(v) \in A$, each link $e \in \varepsilon$ belongs to a particular relation $\phi(e) \in R$. The object type satisfies $|A| > 1$ or the link type satisfies $|R| > 1$.

Example 1. Figure 1 c) contains two types of relations: social relations and rating relations. Figure 1 c) contains two types of nodes: the user and the item. Specifically, User1 and User2 are connected by social relations. User1 and Item1 are connected by rating relation.

2.2 Matrix Factorization

The basic assumption of matrix factorization (MF) [1] is that a set of k-dimensional features can represent user preferences and item attributes. MF extracts users' and items' latent factor vectors from the rating matrix.

MF randomly initializes the features of users and items into k-dimensional vectors, which are called user (item) latent feature vectors, respectively. Second, the latent feature vector is optimized by the existing labeled data (ratings). To be specific, the inner product of the user's latent factor vectors and the item's latent factor vector is the predicted rating. That is, the consistency of the latent factor vectors

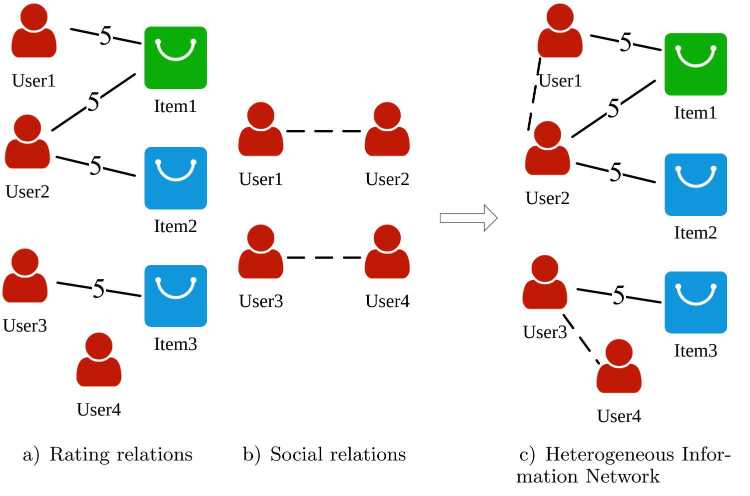


Figure 1. An example of Heterogeneous Information Network

between users and items leads to a predicted rating for the recommendation. The latent feature vector is optimized by minimizing the error between the inner product (predicted ratings) and the labels (real rating). Through iteration, the latent feature vectors of the trained users and items are obtained. Finally, the predicted rating is calculated by the inner product of the latent feature vectors of user i and item j. The method can be expressed as follows:

$$\hat{r}_{ij} = u_i {v_j}^T, \tag{1}$$

where u_i is the latent factor of user i, v_j is the latent factor of item j, and the inner product $\hat{r}_{ij}$ is the predicted rating of user i to item j.

The basic form of matrix factorization is shown in Equation (2). For the rating matrix $\mathbf{R} \in R^{m \times n}$, it can be factorized into a user latent factor matrix, an item latent factor matrix, where k $\ll \min(m, n)$.

$$\mathbf{R} \approx \mathbf{U}\mathbf{V}^T. \tag{2}$$

The optimization objective is shown in Equation (3):

$$\min_{u_i, v_j} \sum_{i,j} \left(r_{ij} - u_i {v_j}^T\right)^2 + \lambda \left(||u_i||^2 + ||v_j||^2\right), \tag{3}$$

where r_{ij} is the observable rating in the rating matrix. The observable rating r_{ij} represents the user's real rating, which can be used as label data. $u_i {v_j}^T$ is the predicted rating of user i to item j. The regularization terms $\lambda(||u_i||^2 + ||v_j||^2)$ are added to Equation (3) to avoid overfitting. By optimizing Equation (3), the error between the predicted rating and observable rating is minimized. To learn the latent factor vectors u_i and v_j, we can solve Equation (3) by gradient descent method.

It can be seen that the predictive accuracy of matrix factorization depends on observable ratings. When the observable ratings decrease, too few samples will affect the training of the latent feature vector. Therefore, in the case of a few ratings, the latent feature vectors cannot effectively represent users or items, which need auxiliary data to improve the prediction accuracy.

3 LOCAL MATRIX FACTORIZATION WITH SOCIAL NETWORK EMBEDDING

In this section, we use auxiliary information to construct sub-matrices more efficiently and present a local matrix factorization model for recommendation. We first introduce the framework of LMFE.

3.1 Framework

As shown in Figure 2, the model can be divided into three parts: network embedding, sub-matrix construction, and sub-matrix factorization. First, we utilize user-item rating relations and user social relations to construct HIN. The network embedding technology is used to extract the representations of the node from HIN, and the representations of the user and the item can be respectively obtained. Then, we randomly select n anchor points (u_t, m_t) in the rating matrix, and use the kernel function to calculate the correlation between the anchor point and the data point. When the correlation is less than the defined width, the data points are classified into sub-matrices built from the anchor points. Finally, we perform the matrix factorization algorithm in the sub-matrix, and the final prediction result can be obtained by the weighted ensemble of the prediction results of the sub-matrix.

In the following sections, we will introduce HIN embedding, sub-matrix construction, and sub-matrix factorization respectively.

3.2 Heterogeneous Information Network Embedding

This section consists of two parts: heterogeneous information network construction and network embedding.

HIN construction. Let $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_m\}$ denote a set of users, and $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$ denote a set of items. $G_1 = (\mathcal{U}, \varepsilon_1)$ denotes the graph constructed from social relations, where $\varepsilon_1 = \{r_1, r_2, \ldots, r_f,\}$. r_f indicates that there is a social relation between users. $G_2 = (\mathcal{U}, \mathcal{I}, \varepsilon_2)$ denotes the graph constructed from user-item rating, where $\varepsilon_2 = \{r_1, r_2, \ldots, r_r\}$. r_f indicates that the

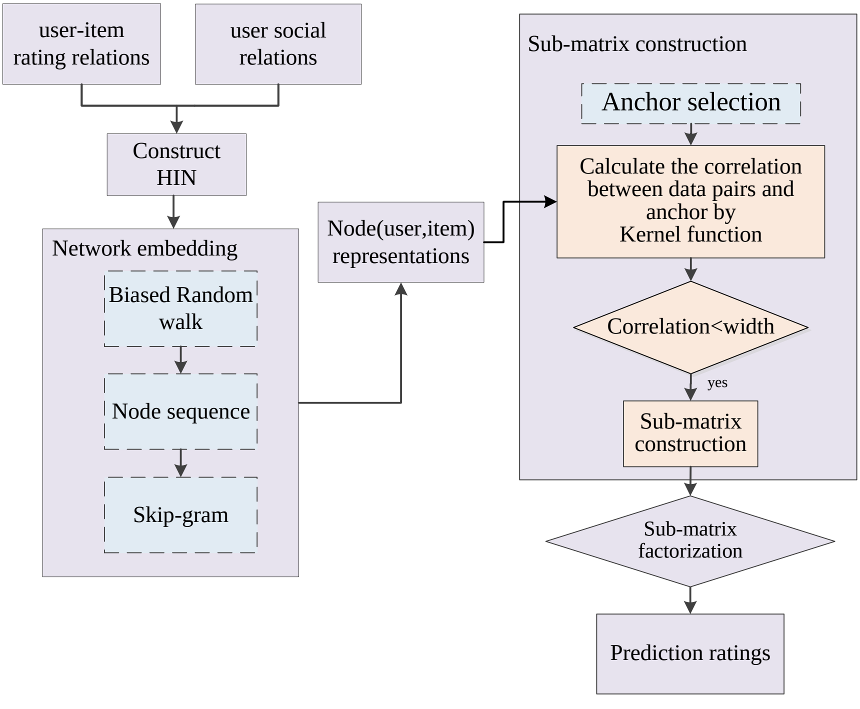


Figure 2. The framework of the LMFE model

user has a rating for the item. Merge the social network G_1 and the rating graph G_2 to construct a HIN $G = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{U}, \mathcal{I} \subset \mathcal{V}$, $\varepsilon_1, \varepsilon_2 \subset \mathcal{E}$.

Network embedding. A network embedding method is designed to learn the representation vectors of users and items in HIN [17, 18, 19]. It needs to sample the node sequence in HIN. Random walk is a classic method of sampling node sequences in homogeneous network. However, there are different types of nodes in a heterogeneous network, such as user and item in Figure 1 c). The random walk cannot be directly applied to a heterogeneous network because a chaotic sequence of nodes is meaningless. Since it is necessary to consider which type of node should be selected as the target node when the random walk starts from the current node [20].

There are two types of nodes in HIN, user and item. We need to formulate a strategy for selecting target nodes. Specifically, when the current node is an item, only the user node is connected to it. Therefore, a user can be selected as the target node with uniform probability. When the current node is a user, there are two types of nodes, user or item, connected to it. First, it needs to determine whether the target node is a user or an item. In the case that the current node is a user, the jump probability of a random walk is formalized as follows:

$$\Pr(\mathcal{V}_{tar} \mid \mathcal{U}_{cur}) = \begin{cases} \alpha \frac{1}{|\mathcal{N}^{user}(\mathcal{U}_{cur})|}, & (\mathcal{U}_{cur}, \mathcal{V}_{tar}) \in \varepsilon_1; \\ (1-\alpha) \frac{1}{|\mathcal{N}^{item}(\mathcal{U}_{cur})|}, & (\mathcal{U}_{cur}, \mathcal{V}_{tar}) \in \varepsilon_2; \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $\mathcal{U}_{cur}$ denotes the current node (user), $\mathcal{V}_{tar}$ is the target node (user or item) where the random walk to jump, Pr denotes the probability distribution of selecting the next node from the current node in the process of random walk. $\alpha \in [0,1]$ is set to control the probability of selecting a user as the target node. $\mathcal{N}^{user}(\mathcal{U}_{cur})$ denotes the neighbor set for the current node with the type of user. $\mathcal{N}^{item}(\mathcal{U}_{cur})$ denotes the neighbor set for the current node with the type of item. It can be seen that, with the increase of α, the probability of selecting the nodes with the type of user by random walk will increase. Biased random walks are performed on the heterogeneous information network G to generate a set of node sequences D. D contains the user node sequence set D_{user} and the item node sequence set D_{item}.

Node sequence filtering. The node sequence contained in D_{user} and D_{item} consists of nodes with different types. Such as the sequence *path*1: "user1, item1, user2", this sequence contains two types of nodes: *user* and *item*. We focus on the similarity of nodes with the same type. Therefore, when constructing user (item) co-occurrence pairs, the node sequence with the same type should be extracted from the node sequence. The node sequence containing only users is extracted from $path \in D_{user}$, and the node sequence containing only items is extracted from $path \in D_{item}$. We update the user node sequences set D_{user} and the item node sequences set D_{item}, respectively.

The node sequence can reflect the co-occurrence probability of nodes. For example, in Figure 1 c), the node sequence *path*1: "user1, user2" reflects the similarity of the two users' preferences for item1; the node sequence *path*2: "user3, user4" exists because the two users have a social relation.

Similar to existing network embedding methods, we utilize skip-gram with negative sampling (SGNS) [21] to learn the representation vectors $e_v \in R^{n \times d}$. In detail, the process can be divided into two steps:

Step 1. Set the sliding window size $w = 2$. From user node sequence set D_{user} and item node sequence set D_{item}, we extract co-occurrence node sequence $\{v_i, v_{i+1}\}_{w=2}$, $\{v_{i-1}, v_i\}_{w=2}$.

Step 2. In the condition of independent hypothesis, set objective function $\Pr(\{v_{i-w}, \dots, v_{i+w}\} \backslash v_i \mid \Phi(v_i)) = \prod_{\substack{j=i-w \\ j \neq i}}^{i+w} \Pr(v_j \mid \Phi(v_i))$. $\Phi(v_i) \in R^d$ represents the representation vector of node v_i. The representation vector $\Phi(v_i) \in R^d$ is trained by maximizing the probability of node co-occurrence Pr.

3.3 Sub-Matrix Construction

We assume the rating matrix is not a global low-rank but a local low-rank. Not all users are closely correlated in the local model, but local users are highly correlated. The process of sub-matrices construction consists of anchor selection and correlation calculation.

Anchor selection. We define a binary group of users and items as a pair of data points (u_t, m_t). We select part of these data points as anchor points, which are the basis for constructing the sub-matrix.

The selection of anchor points is the basis for the construction of a sub-matrix. We adopt three anchor selection methods, namely randomly selecting, selecting anchor points in the test set, and selecting in the training set. According to the above methods, q data points (u_t, m_t) are selected from the rating matrix.

Correlation calculation. The correlation between anchor points and data points is calculated to determine whether data points are classified as submatrices belonging to anchor points. The correlation is determined by user correlation and items correlation, the calculation method is as Equation (5). The correlation between the users (items) refers to the cosine distance calculation of the user (item) representation vector. We use Equation (6) to calculate the correlation. When the cosine distance is less than the threshold, we calculate its cosine distance as the correlation; otherwise, the correlation is set to 0.

$$E(d_i, a_t) = E_h(u_i, u_t) \times E_h(m_i, m_t). \tag{5}$$

The left half of Equation (5) $E(d_i, a_t)$ refers to the correlation between the data point $d_i = (u_i, m_i)$ and anchor point $a_t = (u_t, m_t)$. The right half of Equation (5) refers to the user and item correlation, which is measured by the Epanechnikov kernel function $E_h(s_1, s_2)$. The calculation method is shown in Equation (6):

$$E_h(s_1, s_2) \propto (1 - d(s_1, s_2))\, 1_{[d(s_1,s_2)<h]}, \tag{6}$$

where s denotes the data point or anchor point. In Equation (7), we use the distance function $d(s_i, s_t)$ to denote the distance between the data point and the anchor point.

$$d(s_i, s_t) = arccos\left(\frac{e_{s_i} e_{s_t}}{||e_{s_i}|| \cdot ||e_{s_t}||}\right), \tag{7}$$

where $e_s i$ denotes the node representation obtained in Section 3.2.

When a larger value of $d(s_i, s_t)$ indicates a larger distance between s_i and s_t, which means the correlation decreases. h denotes the threshold. If $d(s_i, s_t) > h$, $E_h(s_1, s_2)$ is 0, indicating that the correlation between s_i and s_t exceeds the threshold, and the correlation is set to 0. $E(d_i, a_t) = 0$ means that the data point d_i

is excluded from the sub matrix based on the anchor point a_t. In particular, there is a situation that the correlation between a data point and all anchor points is 0. The data point will be constructed into the sub-matrix based on the anchor, which has the highest correlation with this data point.

In particular, if the correlation between the data point and any anchor point does not meet the threshold. We add it into the sub-matrix constructed by the anchor point with the highest correlation.

Algorithm 1 Sub-matrix construction
Input: rating matrix R_{mn}, node representation e_v;

```
Randomly select q anchor points (u_t, m_t)
for t = 1, 2, ..., q do
   for i = 1, 2, ..., m do
      if d(u_i, u_t) < h do
         U_t ← U_t ∪ u_i //update the user set of the sub-matrix
   for j = 1, 2, ..., n do
      if d(m_i, m_t) < h do
         M_t ← M_t ∪ m_i //update the item set of the sub-matrix
   return sub-matrix R^t, user ∈ U_t, item ∈ M_t
```

3.4 Sub-Matrix Factorization

In this section, we perform matrix factorization on the sub-matrix obtained in Section 3.2 to predict the ratings. In particular, a data point can belong to multiple sub-matrices. The predicted ratings in multiple sub-matrices were weighted to obtain the ratings $\hat{r}_{ij}$ of the LMFE model. The specific calculation method is shown in Equation (8).

$$\hat{r}_{ij} = \sum_{t=1}^{q} u_i^t \left(v_j^t\right)^T \frac{w_{ij}^t}{\sum_{s=1}^{q} w_{ij}^s}, \tag{8}$$

where $w_{ij}^t = E(u_i, u_t) \times E(m_j, m_t)$ denotes the correlation between $d_i = (u_i, m_j)$ and the anchor point $a_t = (u_t, m_t)$. $\sum_{s=1}^{q} w_{ij}^s$ is the sum of the similarity between data point $d_i = (u_i, m_j)$ and all anchor points. $\frac{w_{ij}^t}{\sum_{s=1}^{q} w_{ij}^s}$ represents the rating weight of the data points $d_i = (u_i, m_j)$ in the sub-matrix R^t. Finally, the prediction rating $\hat{r}_{ij}$ is obtained by weighting the prediction rating $u_i^t(v_j^t)^T$ of all sub-matrices. $||u_i^t||^2$, $||v_j^t||^2$ denotes the regularization term, λ denotes the regularization coefficient, which is used to prevent the model from overfitting.

The prediction model is trained by minimizing the error between the predicted rating and the observed rating. The objective function is as follows:

$$\min_{u_i, v_j} \left(r_{ij} - \sum_{t=1}^{q} u_i^t \left(v_j^t\right)^T \frac{w_{ij}^t}{\sum_{s=1}^{q} w_{ij}^s} + \lambda \left(||u_i^t||^2 + ||v_j^t||^2\right) \right). \tag{9}$$

4 EXPERIMENTS

In this section, the proposed LMFE method is verified by experiments on two real datasets and compared with the existing methods.

4.1 Evaluation Metric

We choose Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics. RMSE and MAE are common metrics for measuring the accuracy of rating prediction. RMSE and MAE are defined as follows:

$$RMSE = \sqrt{\frac{1}{|D_{test}|} \sum_{(i,j) \in D_{test}} \left(r_{i,j} - \hat{r}_{i,j}\right)^2}, \tag{10}$$

$$MAE = \frac{1}{|D_{test}|} \sum_{(i,j) \in D_{test}} |r_{i,j} - \hat{r}_{i,j}|, \tag{11}$$

where D_{test} denotes the test set, $r_{i,j}$ indicates the observed rating (the label or real ratings), $\hat{r_{i,j}}$ denotes the predicted rating. RMSE and MAE represent the error between the observed and predicted ratings, which means that the smaller the values of RMSE and MAE, the better the model's performance.

4.2 Experimental Environment and Settings

The experiments are implemented with Python 2.7 and tested on a server with a 3.10 GHz Intel Core i5-2400 CPU, 64 GB RAM, and Windows 10 professional x64. Specifically, we used packages such as Keras, and NumPy for model training and rating prediction.

The algorithm is validated on the Yelp and Douban dataset. Yelp is an American review site where users can rate items and maintain social relations between users. The platform also keeps the user's social relations. The dataset contains 200 000 ratings of 22 496 items from 37 000 users on a scale of 1 to 5. The dataset contains 140 345 user social relations. This dataset was provided by the Yelp Dataset Challenge.

Douban Movie is a movie community website where users can rate movies they have seen. The website also saves the social relations between users. Specifically, the dataset includes 3 030 user ratings of 6 971 movies from 3 022 users on a scale of 1 to 5. The data set contains 1366 social relationship information and 195 493 movie ratings. This dataset was provided by Douban.

The statistics for the two datasets are shown in Table 1. In the table, rating density represents the density of the rating matrix, and the social edges represent the social relations.

From the statistics of these two datasets, it can be seen that the scale of the Yelp dataset (37 000 users, 22 500 items) is larger than that of Douban. On the other hand, in the Yelp dataset, the rating data is sparser (Rating Density = 0.023 %). By comparing the experimental results of these two datasets, we can observe scenarios where the algorithm is more applicable. We divided the dataset into a training set and a test set in a ratio of 8 : 2. Five experiments were repeated, and the average RMSE and MAE were taken as experimental results.

Datasets	Users	Items	Ratings	Rating Density	Social Edges
Yelp	37 000	22 500	200 000	0.023 %	140 345
Douban	3 030	7 000	195 493	0.92 %	1 366

Table 1. Statistics of datasets

4.3 Baseline Models

We compare the following methods to our approach:

- RegSVD [22]: The Regularized Singular Value Decomposition model uses only the rating matrix as input of the matrix factorization.
- LLORMA [11]: A local low-rank matrix approximation model. After dividing the rating matrix into sub-matrices, the matrix factorization algorithm is performed on the sub-matrix.
- SocReg [23]: A matrix factorization framework with social regularization. It is a collaborative filtering method and adds user social information as a regular term to the matrix factorization model.
- SLOMA [12]: Collaborative Filtering with Social Local Models. A local matrix factorization method constructing sub-matrices according to the social homophily theory.
- LMFE: The model LMFE proposed in this paper first learns the representation of network nodes from HIN and uses these representations to construct the sub-matrix. Finally, matrix factorization is performed on the sub-matrix to obtain the recommendation result.

These baselines include the classic matrix factorization algorithm RegSVD, which only uses ratings as the basis for recommendation. The local matrix factorization algorithm LLORMA is used to verify the validity of the sub-matrices constructed by LMFE, which is suitable for group recommendation, but it does not use any auxiliary data to improve the prediction accuracy of RS. The social recommendation algorithm SocReg adds auxiliary data to improve the accuracy of recommender systems. SLOMA is a local matrix factorization algorithm with excellent performance. By comparing with SLOMA, we can observe the performance of LMFE.

4.4 Recommendation Effectiveness

The inputs for the algorithms listed in Table 2 are different. The classic matrix factorization algorithm RegSVD and local low-rank algorithm LLORMA only use the rating matrix as the input. SocReg and LMFE add user social relations as the input. As can be seen from Table 2, the prediction error of LMFE (MAE, RMSE) is smaller than that of RegSVD and LLORMA, which proves the effectiveness of joining social relations to construct a sub-matrix.

Datasets	Metrics	RegSVD	LLORMA	SocReg	SLOMA	LARec
Yelp	MAE	1.6277	1.3817	1.3231	1.3082	1.2922
	Improve	20.61 %	6.47 %	2.33 %	1.22 %	
	RMSE	1.7317	1.5385	1.4613	1.3691	1.3341
	Improve	22.96 %	13.28 %	8.70 %	2.55 %	
Douban	MAE	0.5831	0.5730	0.5715	0.5705	0.5505
	Improve	5.59 %	3.93 %	3.67 %	3.50 %	
	RMSE	0.7410	0.7287	0.7250	0.7185	0.6961
	Improve	6.06 %	4.47 %	3.99 %	3.12 %	

Table 2. Performance of different methods, the number of sub-matrices is 30

The prediction accuracy of LMFE was significantly better than that of RegSVD. This indicates that adding auxiliary information to the model can improve the accuracy of rating prediction. On the other hand, the correctness of the local low-rank hypothesis is demonstrated, and the performance of RS is improved by following this hypothesis.

Both LLORMA and LMFE follow the assumption of the local low-rank, but LMFE performs better than LLORMA. This is because LMFE adds auxiliary data to the model. From the statistical characteristics of the dataset, it can be found that the Rating Density in Yelp is 0.023 %, which means that users' rating behavior is scarce. Therefore, it is not enough to use only rating data. Better results can be achieved by using auxiliary data for the recommendation. Both LMFE and SocReg make use of auxiliary data and achieve better performance. However, LMFE uses network embedding technology and auxiliary data to obtain more accurate user and item features. At the same time, better recommendation performance is obtained based on the assumption of the local low-rank.

SLOMA is an effective local matrix factorization method, which adds social relations to the recommender system. SLOMA performs better than RegSVD, LLORMA, and SocReg. This is because the sub-matrix constructed through social relations is a low-rank. Matrix factorization on the low-rank sub-matrices can achieve better performance. Nevertheless, in the recommender system, the correlation between users can be reflected not only through social relations but also through user-item rating relations. Therefore, a heterogeneous information network is utilized to model multiple types of relations and get more accurate user and item representation vectors, which lead a better performance in LMFE.

4.5 Impact of the Number of Local Models

In this section, we studied how performance varies with different numbers of local models (sub-matrices) in LLORMA, SLOMA, and LMFE. We select different numbers of sub-matrices to conduct experiments.

As can be seen from Figure 3, LLORMA, SLOMA, and LMFE achieve a higher RMSE when the number of sub-matrices is 1, which means that the performance of the model is worse. The reason that the number of sub-matrices is 1 means that only one anchor point is selected. In this case, all users are affected by a single anchor user, which is unreasonable. When the number of sub-matrices exceeds 5, the local low-rank rating matrix is processed into multiple low-rank matrices. As a result, the performance of the LLORMA, SLOMA, and LMFE algorithms is gradually improved. As the number of local sub-matrices increases, RMSE of LMFE decreases, and RMSE values tend to be stable when the number of sub-matrices exceeds 35.

Anchor Point	Metrics	LLORMA	SLOMA	LMFE
Anchor point selected	MAE	1.3821	1.3047	1.2910
from the Training set	RMSE	1.5362	1.3685	1.3340
Anchor point selected	MAE	1.3829	1.3058	1.2921
from the Test set	RMSE	1.5331	1.3659	1.3342

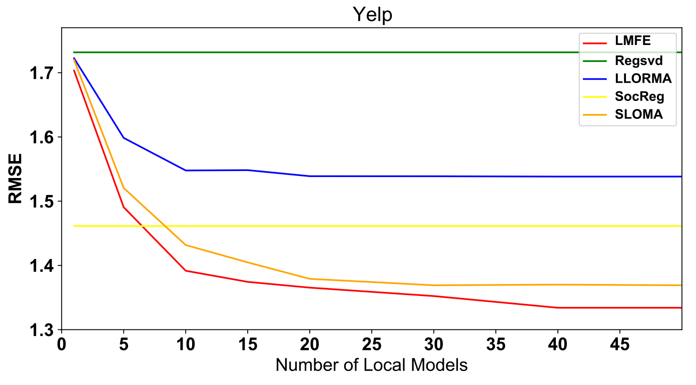
Table 3. Effects of the anchor point on the performance of model (Yelp)

It can be seen that on the Douban dataset, the number of sub-matrices converges faster. Compared with the Yelp dataset, the Douban dataset has a smaller data size. Therefore, by constructing 5 sub-matrices, the rating matrix can be effectively divided. On the Yelp dataset, LMFE has achieved a more significant performance improvement compared to baselines. It also shows that LMFE is more suitable for sparse data environments.

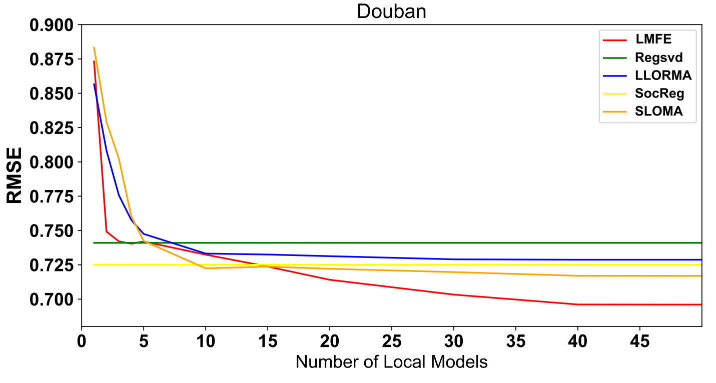
On the other hand, we tested the anchor points selected in the training set and the test set respectively to construct the submatrix. The results show that selecting the anchor points from the training set or the test set has no significant effect on the accuracy of the rating prediction.

4.6 Impact of the Threshold h

For the proposed approach, an important parameter to control the sub-matrices construction is the threshold h. In this section, we discuss the effect of the parameter h on the performance of the model. On the Yelp and Douban datasets, we performed experiments on different values of h, where the range of the threshold h was set to



a) Yelp



b) Douban

Figure 3. Experimental results with different local models

a range of (0.2, 0.4, 0.6, 0.8). Corresponding to each threshold, we select a different number of anchors (sub-matrices) for the experiments. Since the representation vectors with different dimensions d will affect the calculation of similarity, we conduct the experiments in two cases ($d = 8$, $d = 32$), respectively.

Results are shown in Figure 4. It can be seen that when the threshold is set to 0.2, the performance of models in all cases is the worst. This is because when the threshold is 0.2, it means that only users with high similarity (similarity > 0.8) can be constructed into one sub-matrix. According to the algorithm, such a threshold causes many data points not to be classified into any sub-matrix, then the data points are constructed into the sub-matrix based on the anchor, which has the highest correlation with this data point. This strict threshold excludes a large number of data points from the sub-matrix, resulting in performance degradation. When the number of anchors is small, this situation will aggravate the performance degrada-

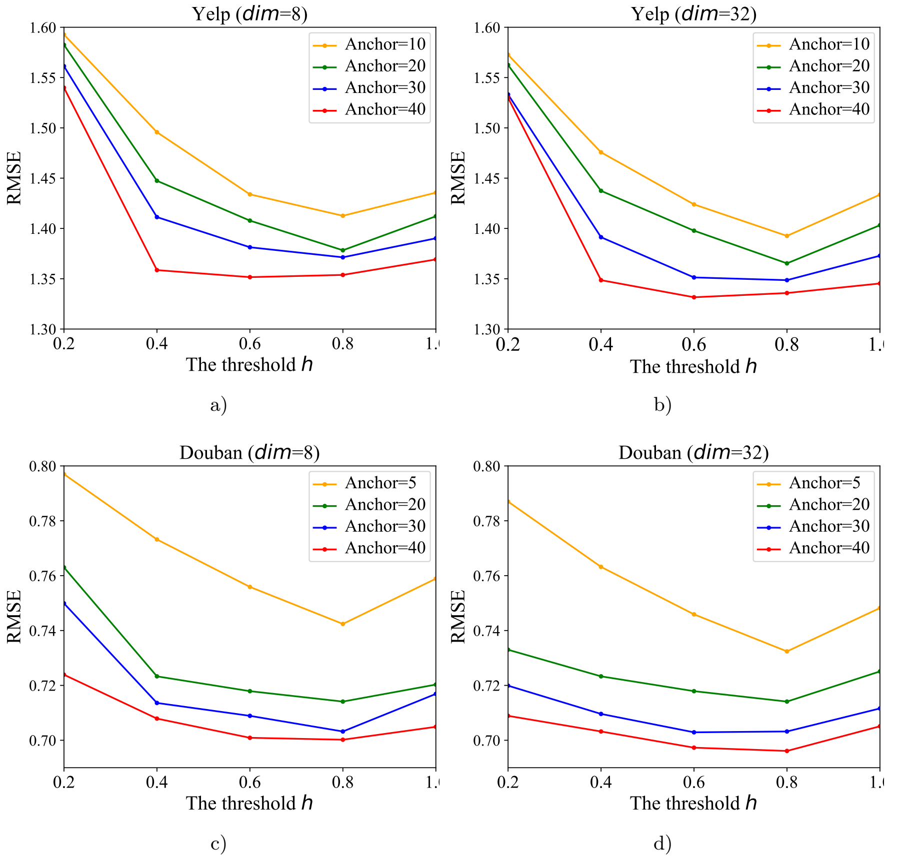
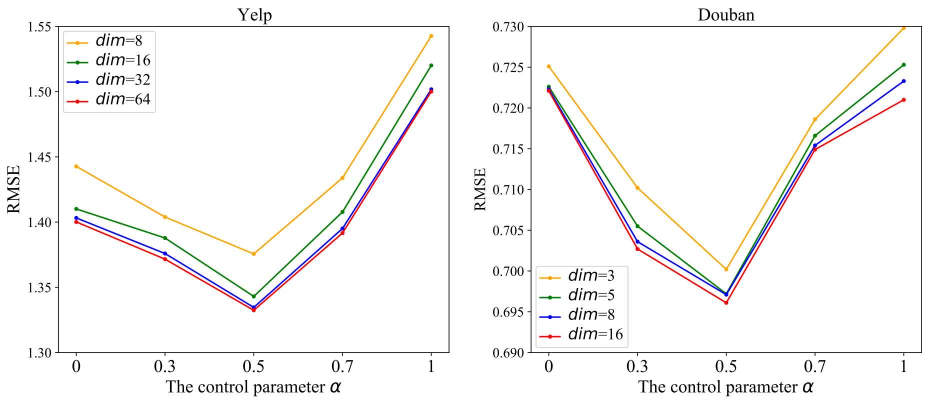


Figure 4. Effects of the threshold h on the performance of the model

tion. The performance becomes better with increasing threshold and then becomes stable. An interesting observation is that on Douban, when $d = 32$ anchor > 20, the performance maintains a steady improvement in the case of threshold changes. The reason is that the Douban dataset is small in scale and the correlation between users (items) is high. When the anchor points are enough, the strict threshold is also enough to make these anchor points cover all users to construct effective sub-matrices. When the threshold is set to 1, it means that each anchor point constructs all data points into a sub-matrix, which is unreasonable. As a result, it leads to a decrease in performance.

4.7 Impact of the Representation Vectors e_v

In this section, we discuss the impact of the representation vectors $e_v \in R^{n\times d}$ on prediction accuracy. The process of representation vector learning includes two important parameters, the control parameter α and the representation vector dimension d. α is set to control the probability of selecting a user as the target node. On the Yelp and Douban datasets, we performed experiments on different values of α, where the control parameter α range was set to $(0, 0.3, 0.5, 0.7, 1.0)$. And the range of the representation vector dimension d was set to a range of $(8, 16, 32, 64)$. For other parameters, the number of local models was 40, the threshold parameter h was 0.8, and the dimension of latent factor vectors k was 10. The experimental results are shown in Figure 5.



a) Yelp b) Douban

Figure 5. Experimental results with different local models

When $\alpha = 0$, it means that the random walk does not jump to the user, so the node sequence consists of the sequence "user-item-user". The representation vector learned from these node sequences characterizes relations that users who "purchase" the same item have similar vectors. When $\alpha = 0.3$, the accuracy of rating prediction has improved. $\alpha = 0.3$ means that the user has a probability of 30 % to jump to the user node, which means that the user's social relations are added to the node sequences. This proves that adding social relations to assist the construction of a sub-matrix helps improve the accuracy of rating prediction. When $\alpha = 1$, user representation vectors are learned entirely from user social relations, resulting in bad performance. Therefore, when modeling user relations, multiple types of data should be considered to represent users more accurately.

It can be seen from the experimental data on the two datasets that the prediction accuracy is improved with the increase of dimension d. On Yelp, when $d = 8$, the dimension of the representation vector is too low to accurately represent users and items, leading to bad prediction performance. The prediction performance of 32-dimensional and 64-dimensional representation vectors are similar, indicating that 32-dimensional representation vectors are sufficient to represent users and items.

On Douban, due to the small data scale, the experiment set a low representation vector dimension ($d = 3, 5, 8, 16$). When the dimension of the representation vector is increased to 5, the prediction accuracy gradually stabilizes, indicating that the 5-dimensional representation vector is enough to represent the users and items.

5 CONCLUSIONS

To solve the problem that the traditional matrix factorization algorithm cannot construct sub-matrix effectively, a network embedding-based local low-rank matrix factorization algorithm LMFE is proposed. LMFE uses social data and rating data to build heterogeneous information networks and then learn user and item representation vectors from them, which can accurately represent user preferences and item attributes. Based on the assumption of local low-rank and the representation vector of user and item, a sub-matrix construction method was proposed. Finally, the prediction rating of the sub-matrix is weighted to obtain the final prediction rating.

Funding Statement

This work was supported in part by the Henan Province Science and Technology Research Project (Nos. 242102210091, and 242102210144) and the Key Research Project plan of Henan Higher Education Institutions (Nos. 24A520011, and 24A520-008).

REFERENCES

[1] Koren, Y.—Bell, R.: Advances in Collaborative Filtering. In: Ricci, F., Rokach, L., Shapira, B. (Eds.): Recommender Systems Handbook. Springer US, Boston, MA, 2015, pp. 77–118, doi: 10.1007/978-1-4899-7637-6_3.

[2] Shi, C.—Hu, B.—Zhao, W. X.—Yu, P. S.: Heterogeneous Information Network Embedding for Recommendation. IEEE Transactions on Knowledge and Data Engineering, Vol. 31, 2019, No. 2, pp. 357–370, doi: 10.1109/TKDE.2018.2833443.

[3] Wang, Z.—Liu, H.—Du, Y.—Wu, Z.—Zhang, X.: Unified Embedding Model over Heterogeneous Information Network for Personalized Recommendation. Proceedings of the 28$^{\text{th}}$ International Joint Conference on Artificial Intelligence (IJCAI '19), AAAI, 2019, pp. 3813–3819, https://www.ijcai.org/proceedings/2019/0529.pdf.

[4] Liu, H.—Jiang, Z.—Song, Y.—Zhang, T.—Wu, Z.: User Preference Modeling Based on Meta Paths and Diversity Regularization in Heterogeneous Information Networks. Knowledge-Based Systems, Vol. 181, 2019, Art. No. 104784, doi: 10.1016/j.knosys.2019.05.027.

[5] Mnih, A.—Salakhutdinov, R. R.: Probabilistic Matrix Factorization. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (Eds.): Advances in Neural Information Processing Systems 20 (NIPS 2007). Curran Associates, Inc., 2007, pp. 1257–1264, `https://proceedings.neurips.cc/paper_files/paper/2007/file/d7322ed717dedf1eb4e6e52a37ea7bcd-Paper.pdf`.

[6] Hong, L.—Doumith, A. S.—Davison, B. D.: Co-Factorization Machines: Modeling User Interests and Predicting Individual Decisions in Twitter. Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13), 2013, pp. 557–566, doi: 10.1145/2433396.2433467.

[7] Koren, Y.—Bell, R.—Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer, Vol. 42, 2009, No. 8, pp. 30–37, doi: 10.1109/MC.2009.263.

[8] Liu, G.—Meng, K.—Ding, J.—Nees, J. P.—Guo, H.—Zhang, X.: An Entity-Association-Based Matrix Factorization Recommendation Algorithm. Computers, Materials & Continua, Vol. 58, 2019, No. 1, doi: 10.32604/cmc.2019.03898.

[9] Modanli, M.—Faraj, B. M.—Ahmed, F. W.: Using Matrix Stability for Variable Telegraph Partial Differential Equation. An International Journal of Optimization and Control: Theories & Applications (IJOCTA), Vol. 10, 2020, No. 2, pp. 237–243, doi: 10.11121/ijocta.01.2020.00870.

[10] Lee, J.—Kim, S.—Lebanon, G.—Singer, Y.—Bengio, S.: LLORMA: Local Low-Rank Matrix Approximation. Journal of Machine Learning Research, Vol. 17, 2016, No. 15, pp. 1–24, `http://jmlr.org/papers/v17/14-301.html`.

[11] Zhang, M.—Hu, B.—Shi, C.—Wang, B.: Local Low-Rank Matrix Approximation with Preference Selection of Anchor Points. Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion), 2017, pp. 1395–1403, doi: 10.1145/3041021.3051148.

[12] Zhao, H.—Yao, Q.—Kwok, J. T.—Lee, D. L.: Collaborative Filtering with Social Local Models. 2017 IEEE International Conference on Data Mining (ICDM), 2017, pp. 645–654, doi: 10.1109/ICDM.2017.74.

[13] Lam, X. N.—Vu, T.—Le, T. D.—Duong, A. D.: Addressing Cold-Start Problem in Recommendation Systems. Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, 2008, pp. 208–211, doi: 10.1145/1352793.1352837.

[14] Yu, J.—Gao, M.—Li, J.—Yin, H.—Liu, H.: Adaptive Implicit Friends Identification over Heterogeneous Network for Social Recommendation. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18), 2018, pp. 357–366, doi: 10.1145/3269206.3271725.

[15] Jamali, M.—Ester, M.: A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10), 2010, pp. 135–142, doi:

10.1145/1864708.1864736.

[16] Sun, Y.—Han, J.: Mining Heterogeneous Information Networks: A Structural Analysis Approach. ACM SIGKDD Explorations Newsletter, Vol. 14, 2013, No. 2, pp. 20–28, doi: 10.1145/2481244.2481248.

[17] Zhao, H.—Yao, Q.—Li, J.—Song, Y.—Lee, D. L.: Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17), 2017, pp. 635–644, doi: 10.1145/3097983.3098063.

[18] Perozzi, B.—Al-Rfou, R.—Skiena, S.: DeepWalk: Online Learning of Social Representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[19] Grover, A.—Leskovec, J.: node2vec: Scalable Feature Learning for Networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[20] Hussein, R.—Yang, D.—Cudré-Mauroux, P.: Are Meta-Paths Necessary?: Revisiting Heterogeneous Graph Embeddings. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18), 2018, pp. 437–446, doi: 10.1145/3269206.3271777.

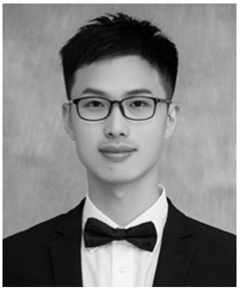
[21] Levy, O.—Goldberg, Y.: Neural Word Embedding as Implicit Matrix Factorization. Advances in Neural Information Processing Systems 27 (NIPS 2014), 2014, https://proceedings.neurips.cc/paper_files/paper/2014/file/feab05aa91085b7a8012516bc3533958-Paper.pdf.

[22] Paterek, A.: Improving Regularized Singular Value Decomposition for Collaborative Filtering. Proceedings of KDD Cup and Workshop (KDDCup.07), 2007, pp. 5–8.

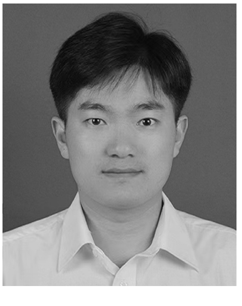
[23] Ma, H.—Zhou, D.—Liu, C.—Lyu, M. R.—King, I.: Recommender Systems with Social Regularization. Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11), 2011, pp. 287–296, doi: 10.1145/1935826.1935877.

Jinmao Xu received his M.Sc. degree from the Zhengzhou University in 2015, and his Ph.D. degree from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2021. His research interests include data mining, machine learning, and recommender systems.

Zhifeng Liu is currently an M.Sc. student at the Zhongyuan University of Technology. His research interests include large language models and artificial intelligence.



Lei Tan received his B.Sc. and M.Sc. degrees from the National University of Defense Technology, Changsha, China, in 2012 and 2014, respectively. He is currently a Lecturer at the PLA Strategic Support Force Information Engineering University. His research interests include steganalysis, machine learning, and recommender systems.



Tianrui Li received his M.Sc. degree from the Tiangong University, Tianjin, China, in 2015 and his Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in 2022. His research interests include network control systems and network security.

Tianqiang Peng received his M.Sc. degree and his Ph.D. degree from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China. His research interests include pattern recognition and intelligent information processing.

Daofu Gong received his B.Sc. degree, his M.Sc. degree, and his Ph.D. degree from the Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2006, 2009, and 2013, respectively. He is currently Associate Professor with the PLA Strategic Support Force Information Engineering University. His research interests include machine learning, network security situation awareness, and social network analysis.