IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 29, 2019, accepted September 11, 2019, date of publication September 25, 2019, date of current version October 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943600

Double Regularization Matrix Factorization Recommendation Algorithm

RUIZHONG DU[1,2], JIAOJIAO LU[1], AND HONGYUN CAI[1,3]
[1]School of Cyber Security and Computer, Hebei University, Baoding 071002, China
[2]Key Laboratory on High Trusted Information System in Hebei Province, Baoding 071002, China
[3]School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

Corresponding author: Jiaojiao Lu (931843520@qq.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61170254, in part by the Hebei Natural Science Foundation Project under Grant F2018201153, and in part by the Hebei Natural Science Foundation Key Program Project under Grant F2019201290.

ABSTRACT With the development of social networks, the research of integrated social information recommendation models has received extensive attention. However, most existing social recommendation models are based on the matrix factorization technique which ignore the impact of the relationships between items on users' interests, resulting in a decline of recommendation accuracy. To solve this problem, this paper proposes a double regularization matrix factorization recommendation algorithm. The algorithm first uses attribute information and manifold learning to calculate similarity. Then, the matrix factorization model is constrained through the regularization of item association relations and user social relations. Experimental results on real datasets show that the proposed method can effectively alleviate problems such as cold start and data sparsity in the recommender system and improve the recommendation accuracy compared with those of existing methods.

INDEX TERMS Social network, recommender system, matrix factorization, item similarity, manifold learning.

I. INTRODUCTION

As an effective method for addressing information overload, the recommender system has become a hot spot of concern in academia and industry. Using a recommendation algorithm, according to user needs, interests, preferences, etc., items that users are interested in are mined from massive data and recommended to users [1]. Traditional recommendation methods can generally be divided into three categories: content-based methods, collaborative filtering methods, and hybrid methods. Collaborative filtering is the most widely used method and mainly includes neighborhood-based collaborative filtering and model-based collaborative filtering [2], [3]. However, traditional recommendation methods face issues related to data sparsity and cold start [4].

Recently, an increasing number of social recommendation [5]–[11] algorithms have used social information from social networks to solve the problems of data sparsity and cold start. Among these algorithms, the socialization recommendation which is based on matrix factorization is the most widely used. Basically, with this method, users with strong social relationships often have similar preferences. Therefore, the introduction of social relationships is conducive to the improvement of the recommendation performance and to a certain extent, and it alleviates problems such as data sparsity in the recommender system. However, in realistic large-scale social network applications, relationships between users are very sparse and constantly changing, and it is difficult to obtain a dense and effective friend or a trust relationship. With the matrix factorization method, the target user's prediction rating is bound by the user's latent feature vector and the item's latent feature vector. Most existing social recommendation models based on matrix factorization focus on the user's friends or trust relationships and have ignored the impact of the relationships between items on the user's preferences. Thus, it is worthwhile to determine a method of incorporating the relationships between items in the recommender system.

To improve the accuracy of recommendation, this paper proposes a double regularization matrix factorization

The associate editor coordinating the review of this manuscript and approving it for publication was Jerry Chun-Wei Lin.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

recommendation algorithm (DRMF). The major technical contributions of this paper are as follows.

(1) First, we introduce a global and local similarity calculation approach based on the known attribute information and a priori information of items, and calculate the overall similarity by integrating the global and local similarities. By using manifold learning, a low-dimensional similarity is obtained according to the overall similarity of the items.
(2) The similarity of the low-dimensional manifold is added to the Pearson correlation calculation method to improve the result and yield a comprehensive item similarity.
(3) Item regularization is included in the social recommendation objective function, and the new objective function can enhance the constraints of the item feature matrix.

II. RELATED WORK

The traditional recommender system ignores the interactions between users. However, relationships in social networks provide rich information that can be used to improve recommendation performance. Social networks have generated rich behavioral interaction information, and many recommendation methods which are based on social networks have been proposed and studied. Social recommendation techniques can be divided into two categories: neighborhood-based methods [6], [7] and model-based methods [8]–[11]. Most neighborhood-based methods either directly or indirectly use social trust to represent the similarity between users. Pal and Jenamani [6] proposed a trust-aware recommendation algorithm based on a user's explicit trust relationships. This method combines user similarity and trust with the collaborative filtering algorithm and achieves good recommendation performance. However, when explicit trust information is difficult to obtain, it affects the accuracy of the recommendation. In view of this situation, Wu *et al.* [7] proposed a neighborhood-based implicit trust recommendation algorithm by using the propagative nature of trust and scale-free complex network properties to limit the range of propagation in the network and improve the accuracy of the neighborhood-based collaborative filtering recommendation algorithm.

Because the matrix factorization model is scalable and flexible, scholars have used it to develop many social recommendation algorithms. These methods can effectively solve the problem of data sparsity. Ma *et al.* [8] proposed the SoReg algorithm, which considers the relationships between friends and trust to vary. The establishment of trust relationship depends on similar interest preferences among users. The establishment of a friendship depends on the social relationships of users in the real world. A regularization recommendation model combining the average ratings of trust users and the ratings of individual trust users is proposed, and the interest propagation between friends can be automatically determined during the learning process, which facilitates interpretability. Guo *et al.* [9] proposed the TrustSVD algorithm, which constrained the user's friend relationships to different degrees and imposed less severe punishments on popular users than on users with lower ratings. The method adds social feedback information based on the original SVD++ model and treats the user's explicit trust relationship and rating information as implicit information; then, it expands the trust relationship information and constructs prediction ratings. Jaehoon *et al.* [10] claimed that the unidirectionality of trust relationships cannot guarantee common characteristics between the trustor and the trustee; they proposed a TCRec clustering algorithm that posits that trustors who follow the same trustee have features in common, based on the assumption that trustors who endorse the same trustee share similar tastes, and user and latent item characteristics were learned so that personalized recommendations could be made.

The above studies all incorporate social relationships into the social recommendation process, which can alleviate the cold-start and data sparsity problems encountered during recommendation to some extent. However, in an actual large-scale social network, the relationships between users are very sparse and constantly changing, and friendship or trust relationships between users cannot be effectively extracted, which affects the accuracy of recommendations.

To further improve the accuracy of recommendations, the relationships between items [12]–[15] were introduced into the recommendation. Liang *et al.* [13] proposed a joint factorization model algorithm that calculates the similarity between items by using the interaction information of cooccurrence matrix points between items. Finally, the weight matrix factorization model is used for recommendation, which improves recommendation quality; however, for this method, there is no constraint on the feature information between items. Xu [14] proposed a recommendation algorithm that combines social information and item relationships based on matrix factorization, imposes regularized constraints on the feature information of items and maintains the stability of data features during matrix factorization and dimension reduction; however, this method considers only the influence of friends' favorite items on the user's choices. Xiao *et al.* [15] proposed the TranSIV generation model, which integrates global information and considers item similarity, and integrated social networking and rating information into a unified model through migration learning to learn user preferences. However, the above methods do not consider the local similarity of items and cannot fully reflect the similarity relationships between items. In addition, the influence of item attribute information on recommendation results is ignored.

III. SOCIAL RECOMMENDATION ALGORITHM WITH ITEM SIMILARITY

A. ITEM SIMILARITY CALCULATION BASED ON ATTRIBUTE FREQUENCY AND ATTRIBUTE AGGREGATION

In the recommendation algorithm, similarity calculation includes cosine similarity, Pearson correlation and modified cosine similarity. In this paper, the Pearson correlation is used to calculate the similarity between item j and item j'. The calculation formula is as follows:

$$Sim(j, j') = \frac{\sum_{p \in P_{jj'}} (R_{pj} - \bar{R}_j) \cdot (R_{pj'} - \bar{R}_{j'})}{\sqrt{\sum_{p \in P_{jj'}} (R_{pj} - \bar{R}_{j'})^2} \cdot \sqrt{\sum_{p \in P_{jj'}} (R_{pj'} - \bar{R}_{j'})^2}} \quad (1)$$

where $P_{jj'}$ is a collection of users who have rated both item j and item j', and R_j and $R_{j'}$ represent the average ratings of item j and item j', respectively. However, when the rating data is sparse, the two items have fewer common ratings, and the item's similarity cannot be accurately calculated. At the same time, using an algorithm strategy that simply relies on the rating to calculate the item's similarity without considering the item's characteristics also affects the recommendation effect to some extent.

Item attribute information can provide additional information to the recommender system and improve the similarity calculation of the item. Therefore, this paper uses item attribute information and prior information to propose the global and local similarity measure of the item to capture its overall similarity. Using the manifold learning method, the low-dimensional similarity is calculated according to the overall similarity of the item, and finally, it is combined with the Pearson correlation calculation method.

Let $O = \{o_1, o_2, \ldots, o_n\}$ denote the item set. Then, the overall similarity between item o_j and item $o_{j'}$ can be calculated as:

$$S(o_j, o_{j'}) = \sum_{a \in A, o_j, o_{j'} \in O} F(LS_a(o_j, o_{j'}), GS_a(o_j, o_{j'})) \quad (2)$$

where $A = \{a_1, a_2, \ldots, a_h\}$ denotes the attribute set, h is the number of item attributes, $LS_a(o_j, o_{j'})$ and $LS_a(o_j, o_{j'})$ indicate the item's global and local similarity constraint functions, respectively, and $F(X, Y)$ is the mapping function of the item's global and local similarity, which indicates the overall similarity of the item.

For global item similarity, we focus on the impact of different values of one attribute on the two terms. First, a set of item values $T_a = \{t_a^{o1}, t_a^{o2}, \ldots, t_a^{on}\}$ is defined, and a frequency counter function $C(t_a^{o_j})$ of a specific value of attribute a of item o_j, which indicates how many times the value $t_a^{o_j}$ (i.e., the value of attribute a of o_j) appears among all items, is introduced. The item global similarity function is defined as equation (3):

$$GS_a(o_j, o_{j'}) = \frac{\frac{C(t_a^{o_j})}{n} \cdot \frac{C(t_a^{o_{j'}})}{n}}{\frac{C(t_a^{o_j})}{n} + \frac{C(t_a^{o_j})}{n} \cdot \frac{C(t_a^{o_{j'}})}{n} + \frac{C(t_a^{o_{j'}})}{n}} \quad (3)$$

where n is the number of items in the entire item space O. As shown in the above function, if the attribute values of the two items appear the same number of times in all items $\frac{n}{2}$, the global similarity is $\frac{1}{5}$, and the method based on the attribute value frequency specifically yields the global similarity of the item.

For the local similarity of the item, the similarity between items o_j and $o_{j'}$ is calculated by aggregating attribute a, and $a' \in A(a' \neq a)$, where $T_{a'} = \{t_{a'}^{o1}, t_{a'}^{o2}, \ldots, t_{a'}^{on}\}$ is a set of values of attribute a'. The conditional probability function is defined as:

$$P_{T_{a'}|t_a^{o_j}}(T_{a'}, t_a^{o_j}) = \sum_{t_{a'}^{o_\varsigma} \in T_{a'}, o_j, o_\varsigma \in O} \frac{C(t_{a'}^{o_\varsigma} \cap t_a^{o_j})}{C(t_a^{o_j})} \quad (4)$$

The local similarity function between items o_j and $o_{j'}$ is defined as:

$$LS_a(o_j, o_{j'}) = \sum_{a', a \in A, a' \neq a} \eta \min(P_{T_{a'}|t_a^{o_j}}(T_{a'}, t_a^{o_j}), P_{T_{a'}|t_a^{o_{j'}}}(T_{a'}, t_a^{o_{j'}})) \quad (5)$$

a' is in attribute set A but not equal to attribute a and η is the weighting factor of each part, where $\eta \in (0, 1)$; $\eta = \frac{1}{n}$ in the model. As shown above, $LS_a(o_j, o_{j'}) \in [0, 1]$.

Finally, the weights of the global similarity and local similarity of the item are assigned. The overall similarity between o_j and $o_{j'}$ is expressed as:

$$S(o_j, o_{j'}) = \sum_{a \in A, o_\iota, o_\tau \in \Omega} \sigma LS_a(o_\iota, o_\tau) + (1 - \sigma) GS_a(o_j, o_{j'}) \quad (6)$$

where σ is the harmonic factor between the global and local similarities of the item, which is used to measure the degree of influence of the global and local similarity. We usually set $\sigma = 0.5$ in this model.

There is a ubiquitous and intrinsic relationship between the items in the recommendation system. Taking Taobao as an example. Each user has rated items by using a discrete number on the scale of 1–5, and items interacted according to categories, manufactures, and prices. The above similarity model can be applied to the recommender systems, and the similarity between the items can be calculated by $S(o_j, o_{j'})$.

Manifold learning is widely used in machine learning and data mining [16]–[19]. The basic idea is to keep the data characteristics in the original space when mapping data from high-dimensional space to low-dimensional space. In this paper, the item manifold learning [20] constraint is added; according to the manifold hypothesis principle, if o_j and $o_{j'}$ are similar in the original data space, then they are similar in the new mapping space. For items o_j and $o_{j'}$, *Knearest* indicates the distance between items. The similarity function in equation (6) is used to calculate the distance between items, as shown in equation (7):

$$Knearest(o_j) = \{o_{j'} | S(o_j, o_{j'}) > S(o_j, o_x), \forall o_x \notin Knearest(o_j), \quad \forall o_{j'} \in Knearest(o_j)\} \quad (7)$$

where $S(o_j, o_x)$ represents the similarity between item j and any item x. For the neighbor of item j, if $S(o_j, o_{j'})$ is greater than $S(o_j, o_x)$, and any item x is not a neighbor of item j, any item j' belongs to item j.

After calculating the distance between the items, then select the K neighbors of the item and calculate the similarity between each item and the neighbor nodes. The low-dimensional similarity of the item is represented by w, and the final similarity expression of the item is obtained:

$$w_{jj'} = \begin{cases} S(o_j, o_{j'}), & o_j \in Knearest(o_{j'}) \vee o_{j'} \in Knearest(o_j) \\ 0, & otherwise \end{cases} \tag{8}$$

The Pearson similarity calculation formula that incorporates the item attribute characteristics is as follows:

$$itemSim(j, j') = \frac{\sum_{j,j' \in O} (R_{pj} - \bar{R}_j) \cdot (R_{pj'} - \bar{R}_{j'}) w_{jj'}}{\sqrt{\sum_{j,j' \in O} (R_{pj} - \bar{R}_{j'})^2} \cdot \sqrt{\sum_{j,j' \in O} (R_{pj'} - \bar{R}_{j'})^2}} \tag{9}$$

B. DOUBLE REGULARIZATION MATRIX FACTORIZATION MODEL

To further predict the missing data of the user-item rating matrix $R_{m\times n}$, the high-dimensional user-item rating matrix is decomposed into a low-dimensional user feature matrix and an item feature matrix. U represents the user feature matrix, and V represents the item feature matrix, as shown in equation (10):

$$R \approx U^T V \tag{10}$$

where $U \in R^{m\times d}$, $V \in R^{n\times d}$, $d < min(m, n)$, and the low rank matrix factorization method approximates the rating matrix according to the products of the d rank factor R. U_i is the latent feature vector of user i, V_j is the latent feature vector of item j, and the predicted rating of user i for item j is expressed as $\hat{R_{ij}} = U_i^T V_j$. The squared loss between the predicted rating and the original rating is used as the loss function, and the loss function is minimized to approximate the rating matrix $R_{m\times n}$.

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 \tag{11}$$

Here, I_{ij} is an index function, indicating that if, user i has rated item j, then it is equal to 1; otherwise, it is equal to 0. To prevent overfitting, two regularization terms are added in formula (11), and the final optimization objective function is calculated as:

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_1}{2} \parallel U \parallel_F^2 + \frac{\lambda_2}{2} \parallel V \parallel_F^2 \tag{12}$$

where $\lambda_1, \lambda_2 > 0$, $\parallel \cdot \parallel_F$ indicates the regularization constraint norm. The stochastic gradient descent method is used to optimize the objective function to obtain the missing value of the original rating.

In social networks, users usually prefer to connect with users who share their interests. Therefore, the user's decision is easily influenced by friends, and the closer the relationship between users, the higher the impact. Adding social relationships based on the above model gradually biases the user's interest toward the preferences of friends in the social network [8]. The objective function is shown in formula (13):

$$\begin{aligned} \min_{U,V} L_1(R, U, V) = {} & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 \\ & + \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{f \in F+(i)} Sim(i, f) \parallel U_i - U_f \parallel_F^2 \\ & + \frac{\lambda_1}{2} \parallel U \parallel_F^2 + \frac{\lambda_2}{2} \parallel V \parallel_F^2 \end{aligned} \tag{13}$$

$\alpha > 0$, $F^+(i)$ is the friend group of the user, $Sim(i, f)$ is the similarity between user i and user f, and different friends have different similarities.

In a real recommendation scenario, because the rating matrix is very sparse, it is not common for friend relationships to have rating records for the same item when calculating user similarity. To alleviate the cold-start and data sparsity problems and improve recommendation performance, this paper incorporates the item regularization constraints, guides the learning process by calculating the similarity between items, and makes similar items easier to recommend to target users, thus, the recommendation results are more interpretable. The item regularization constraint function is given by equation (14):

$$\frac{\beta}{2} \sum_{j=1}^{n} \sum_{j'=1}^{n} itemSim(j, j') \parallel V_j - V_{j'} \parallel_F^2 \tag{14}$$

V_j and $V_{j'}$ represent the latent feature vector of the item, and β is the item regularization constraint parameter used to constrain the item characteristics. The above model uses the regularization method to maintain the stability of the features of the item after dimension reduction, that is, the greater the similarity between the items, the smaller the distance of the corresponding feature vector. Finally, the objective function is given by equation (15):

$$\begin{aligned} \min_{U,V} L_2(R, U, V) = {} & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}(R_{ij} - U_i^T V_j)^2 \\ & + \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{f \in F+(i)} Sim(i, f) \parallel U_i - U_f \parallel_F^2 \\ & + \frac{\beta}{2} \sum_{j=1}^{n} \sum_{j'=1}^{n} itemSim(j, j') \parallel V_j - V_{j'} \parallel_F^2 \\ & + \frac{\lambda_1}{2} \parallel U \parallel_F^2 + \frac{\lambda_2}{2} \parallel V \parallel_F^2 \end{aligned} \tag{15}$$

The user's latent feature matrix by matrix factorization can reflect the similarity of user friend preferences in the social network. The above objective function is optimized based on the similarity between social friends, and the feature vector of the target user is constrained by the user's friends;

TABLE 1. Data description.

Dataset	Number of users	Number of items	Number of ratings	Number of social relations
Epinions	41365	145987	672694	496322
Douban	55296	19896	423690	673489
Ciao	15156	14985	69587	39254
Flister	1845	2267	38561	2029

TABLE 2. Performance comparison on all users.

Dataset	Dimension	Metrics	PMF	SoReg	SVD++	TrustSVD	TCRec	DRMF
Epinions	d=5	MAE	0.987	0.954	0.833	0.826	0.807	0.785
		RMSE	1.298	1.253	1.078	1.065	1.051	1.031
	d=10	MAE	0.917	0.904	0.834	0.826	0.806	0.784
		RMSE	1.205	1.175	1.090	1.065	1.046	1.025
Douban	d=5	MAE	0.822	0.782	0.778	0.752	0.736	0.715
		RMSE	1.084	1.025	1.002	0.998	0.959	0.935
	d=10	MAE	0.814	0.794	0.792	0.764	0.726	0.704
		RMSE	1.071	1.032	1.001	0.982	0.955	0.932
Ciao	d=5	MAE	1.113	0.928	0.771	0.743	0.726	0.707
		RMSE	1.456	1.247	0.985	0.982	0.961	0.939
	d=10	MAE	0.911	0.862	0.757	0.741	0.725	0.706
		RMSE	1.187	1.175	1.005	0.975	0.959	0.936
Flixster	d=5	MAE	0.722	0.682	0.686	0.621	0.616	0.599
		RMSE	0.957	0.886	0.895	0.812	0.796	0.778
	d=10	MAE	0.743	0.676	0.689	0.619	0.615	0.596
		RMSE	0.976	0.883	0.892	0.811	0.794	0.775

this influence is reflected in the social relationship regularity. The regularization term of the item constrains the latent feature matrix of the item and optimizes the similarity between items. The feature vector of the item is constrained by the manifold relationship and its own special attribute. This effect is reflected in the item regularity term.

Finally, the rating matrix, social relationship and item similarity are unified in the matrix factorization framework so that the learned rating matrix is closer to the true value. In this paper, we use the stochastic gradient descent method to optimize the objective function, calculate the new objective function with the new variable value, and iterate until the loss function converges to obtain the final prediction rating.

Step 1: The loss function L_2 is used to bias the variable U_i

$$\frac{\partial_{L_2}}{\partial_{U_i}} = \sum_{j=1}^{n} I_{ij}(U_i^T V_j - R_{ij})V_j + \alpha \sum_{f \in F^+(i)} Sim(i,f)(U_i - U_f) + \alpha \sum_{g \in F^-(i)} Sim(i,g)(U_i - U_g) + \lambda_1 U_i \tag{16}$$

The loss function L_2 is used to bias the variable V_j

$$\frac{\partial_{L_2}}{\partial_{V_j}} = \sum_{i=1}^{m} I_{ij}(U_i^T V_j - R_{ij})U_j + \beta \sum_{j'=1}^{n} itemSim(j,j')(V_j - V_{j'}) + \lambda_2 V_j \tag{17}$$

Step 2: The variables in the model are updated

$$U_i \leftarrow U_i - \omega \frac{\partial_{L_2}}{\partial_{U_i}} \tag{18}$$

$$V_j \leftarrow V_j - \omega \frac{\partial_{L_2}}{\partial_{V_j}} \tag{19}$$

Finally, the user and item latent feature vectors U_i and V_j are obtained separately, and the inner product of U_i and V_j is used to predict the missing value of the original rating matrix R:

$$\hat{R_{ij}} = U_i^T V_j \tag{20}$$

The details of the double regularization matrix factorization recommendation algorithm are as follows:

The time complexity of DRMF is mainly calculated according to the objective function L_2 and the corresponding partial derivative. The time complexity of calculating the objective function L_2 is $O(d|R| + d|F| + mdh)$, where $|R|$ and $|F|$ represent the number of ratings and the number of social relationships, respectively, d represents the potential feature dimension, h indicates the number of attributes, and m represents the number of users. The time complexity for calculating partial derivatives is $O(d|R|)$ and $O(mdh)$. The time complexity of this algorithm is $O(d|R|\bar{r} + d|F| + mdh)$, where $\bar{r}$ represents the average number of ratings for users. Therefore, the overall complexity of the algorithm is linear with the number of ratings, the number of social relationships, the number of project attributes, and the number of users, and the algorithm can handle large-scale data.

Algorithm 1

Input: A_o: item attribute in item space O; F: social relationship set; α: social relationship constraint; β: item relationship constraint; λ_1: user regularization parameter; λ_2: item regularization parameter; d: latent feature vector dimension; ω:learning rate;Max: maximum number of iterations; ϖ:stop the iteration threshold ϖ.
Output: U, V.
Randomly initialize U and V
where oj and $oj' \in O$, Eq.(6) // Construct the item relationship relationship
where o_j and $o_{j'} \in O, Eq.(7)$ // Construct a neighborhood relationship
Calculate the low-level similarity of the item, $Eq.$ (8)
Calculate the similarity of the attributes of the fusion item, $Eq.$ (9)
Calculate the objective function, $Eq.$ (15)
while $max < Max$ or $L_{max} - L_{max-1} < \varpi$ do
  for $p \in [1, m]$ do
  for $o \in [1, n]$ do
    if $S(o_j, o_{j'}) > 0$ then
      $U_i \leftarrow U_i - \omega \frac{\partial_{L_2}}{\partial_{U_i}}$ // Update U_i
      $V_j \leftarrow V_j - \omega \frac{\partial_{L_2}}{\partial_{V_j}}$ // Update V_j
    end if
  end for
  end for
end while
return U, V

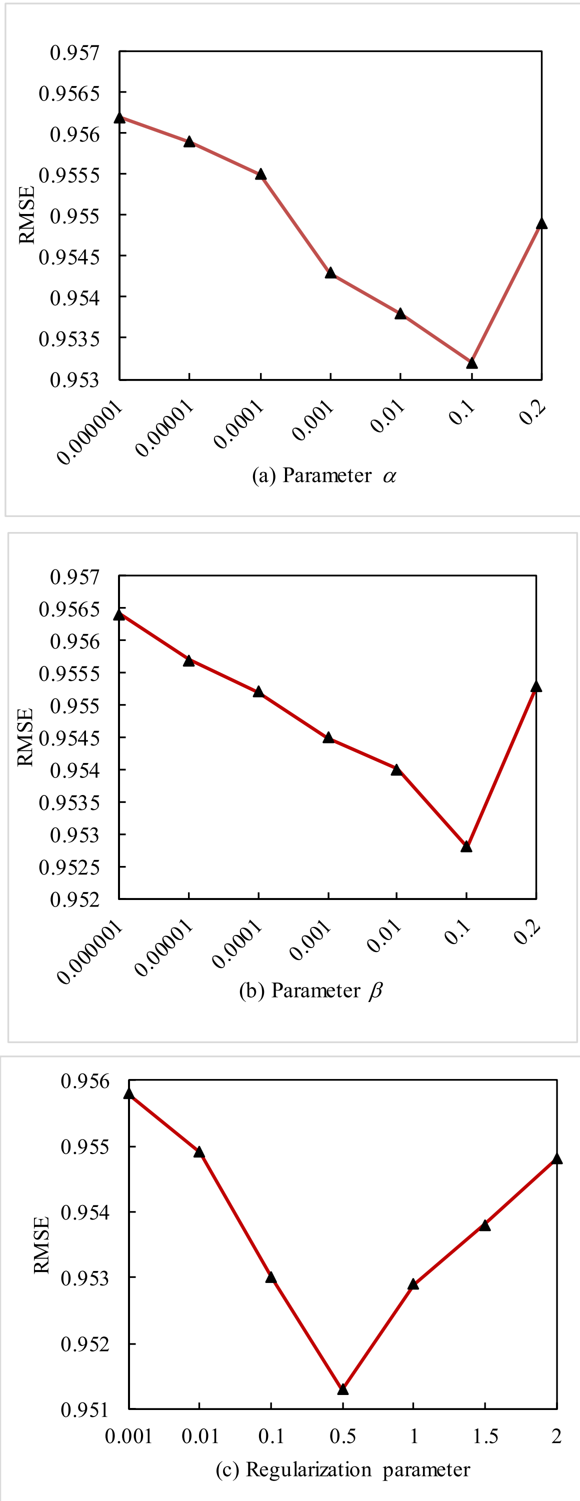
IV. EXPERIMENT

A. DATASET

To analyze the influences of different types of information on the recommendation results, four datasets containing social relations and rating information, i.e., Epinions, Douban, Ciao, and Flixster, are selected to verify the proposed algorithm.

Epinions is an item review site that allows users to rate items; including item categories, prices, descriptions, etc., as well as user social relationships. Douban is a social networking site that allows users to rate movies, music, etc.; the properties of a movie include movie ID, movie name, and category, and the user can add friends to build a social relationship. Ciao is an item review site that allows users to rate items; item attributes include category, name, and ID. Flixster is a movie site that allows users to rate movies; movie attributes include movie name and category. The details of the four datasets are shown in Table 1.

In this paper, the five-fold cross-validation method is used to train and test the recommended model, and the final test results are taken as the mean of five experimental results.



(a) Parameter α
(b) Parameter β
(c) Regularization parameter

FIGURE 1. Impact of parameters.

B. COMPARISON METHOD AND EVALUATION METRICS

To verify the validity of the DRMF algorithm, it is compared with five representative algorithms. The comparison algorithms are as follows:

TABLE 3. Performance comparison on cold start users.

Dataset	Dimension	Metrics	PMF	SoReg	SVD++	TrustSVD	TCRec	DRMF
Epinions	d=5	MAE	1.459	1.406	0.872	0.897	0.856	0.833
		RMSE	1.778	1.743	1.138	1.121	1.086	1.067
	d=10	MAE	1.161	0.994	0.865	0.861	0.855	0.831
		RMSE	1.440	1.225	1.135	1.114	1.081	1.059
Douban	d=5	MAE	1.105	1.066	0.889	0.876	0.851	0.830
		RMSE	1.398	1.366	1.111	1.082	1.061	1.039
	d=10	MAE	0.959	0.954	0.892	0.877	0.851	0.828
		RMSE	1.214	1.209	1.110	1.101	1.060	1.038
Ciao	d=5	MAE	1.414	0.913	0.765	0.727	0.704	0.681
		RMSE	1.771	1.114	0.972	0.966	0.931	0.913
	d=10	MAE	1.055	0.765	0.740	0.725	0.703	0.679
		RMSE	1.348	1.087	0.969	0.964	0.931	0.913
Flixster	d=5	MAE	0.822	0.791	0.705	0.685	0.657	0.627
		RMSE	1.087	0.988	0.924	0.905	0.846	0.815
	d=10	MAE	0.775	0.769	0.688	0.678	0.653	0.623
		RMSE	1.017	0.966	0.915	0.893	0.849	0.816

(1) PMF [2]: probability-based matrix factorization recommendation algorithm proposed by Salakhutdinov *et al.*

(2) SoReg [8]: a social regularization recommendation algorithm based on social relations proposed by Ma *et al.*

(3) SVD++ [3]: a recommendation algorithm proposed by Koren *et al.* that simultaneously considers user information and item information

(4) TrustSVD [9]: the extended SVD++ model proposed by Guo *et al.* that introduces social information

(5) TCRec [10]: method proposed by Jaehoon *et al.* in which trustors who follow the same trustee have features in common

According to the relevant literature [8]–[10], the regularization parameters of PMF, SoReg, and the SVD++ algorithm are $\lambda_u = \lambda_v = 0.001$; for SoReg, $\beta = 0.1$. For TrustSVD, $\lambda_u = \lambda_v = 1.2$, and $\lambda_t = 0.9$. For TCRec, $\lambda_u = \lambda_v = \lambda_s = 0.1$, $\lambda_b = 0.01$, $\beta = 100$.

In this paper, the performance of the recommendation algorithm is measured according to two evaluation metrics: the root mean square error (RMSE) and the mean absolute error (MAE). The accuracy of the results is determined by calculating the error between the true rating and the predicted rating. The smaller the error, the higher the accuracy of the recommendation. The evaluation metrics are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i,j}(R_{ij} - \hat{R_{ij}})^2}{T}} \tag{21}$$

$$MAE = \frac{\sum_{i,j}|R_{ij} - \hat{R_{ij}}|}{T} \tag{22}$$

where T represents the number of ratings in the test set, R_{ij} represents the true user's rating on the item, and $\hat{R_{ij}}$ indicates the predicted rating.

C. EXPERIMENTAL PARAMETERS SETTING

In order to explore the influence of parameters in the DRMF algorithm on the recommendation results. Taking the Douban data set as an example, an experiment is carried out with 5 dimensions of latent features. According to the literature [5], the parameters α and β are set to 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, and 0.2. Parameter $\lambda_u = \lambda_v$ is set to 0.001, 0.01, 0.1, 0.5, 1, 1.5, and 2.

TABLE 4. Distribution of users with different ratings.

Dataset	[0,10]	[11,30]	[31,50]	[51,90]	[91,+∞]
Epioions	63.75%	23.27%	7.34%	5.43%	0.21%
Douban	81.22%	11.41%	5.27%	1.69%	0.41%

Here, α controls the influence of the user's social relationship on the recommendation result, β controls the impact of the regularization relationship of the item on the recommendation result, and λ_u and λ_v represent the constraint parameters of the user and the item, respectively.

For α and β, it can be seen from Figure 1 (a) and (b) that, as the values of α and β increase, the RMSE first decreases and then increases. Obviously, the optimum values of α and β are 0.1. Therefore, the regularization of the item and social relationship have a certain impact on the learning process, which indicates that combining item similarity and regularization of the social recommendation model improves the recommendation accuracy of the recommender system.

For λ_u and λ_v, it can be seen from (c) of Figure 1 that the optimal values of λ_u and λ_v (i.e., the values for which the point error of the algorithm is the smallest) are 0.5. When the values of λ_u and λ_v are either too large or too small, the results are unsatisfactory. Therefore, for the user and the item's regularization parameters, $\lambda_u = \lambda_v = 0.5$ is the best value.

D. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the prediction accuracy of the DRMF algorithm, it compares with five algorithms PMF, SoReg, SVD++, TrustSVD, and TCRec.

First, we conducted experiments and compared different feature dimensions. This experiment compared the results of various algorithms using the Epinions, Douban, Ciao, and Flixster datasets. The latent feature dimension was set

TABLE 5. Distribution of users with different social relationships.

Dataset	[0,5]	[6,10]	[11,20]	[21,50]	[51,90]	[91,500]	[501, +∞]
Epioions	62.32%	11.38%	9.26%	6.52%	6.37%	3.28%	0.87%
Douban	24.17%	21.25%	18.37%	16.42%	14.27%	5.52%	0

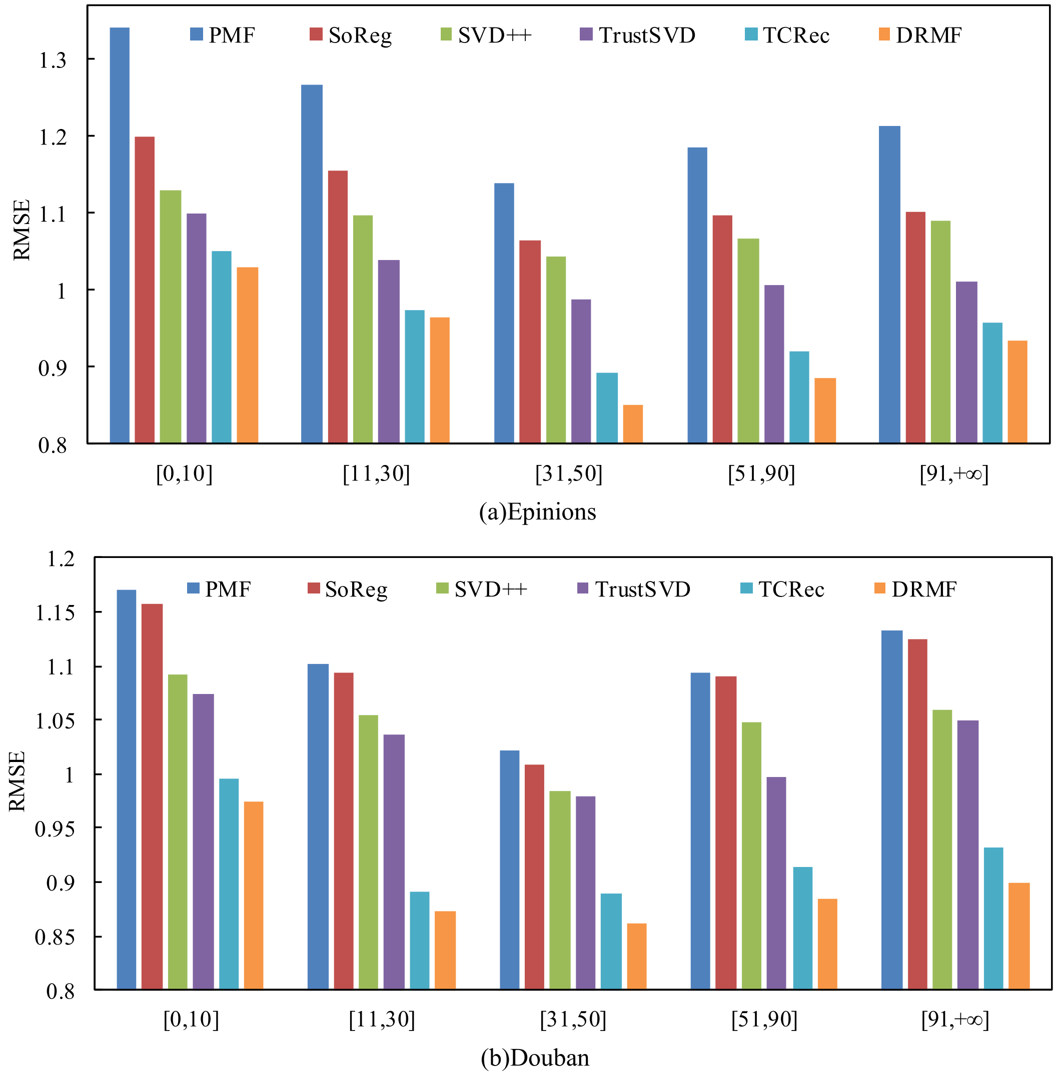


FIGURE 2. Comparison of RMSE for six algorithms.

to d = 5 and 10 to verify the accuracies of various algorithms. Similar to reference [8], users with less than 5 ratings in the training set are called cold-start users. The experimental results of six recommended algorithms are shown in Table 2 and Table 3. Table 2 shows the comparison results for the overall user set, and Table 3 shows the comparison results for the cold start user set. It can be seen from Table 2 that, compared with PMF, the recommendation accuracy of SVD++ is significantly improved for the overall user set, indicating that user information and item information can improve the prediction accuracy of the recommendation algorithm. As seen from Table 3, in the cold start user set, DRMF is compared with several other experimental methods, and the recommendation accuracy is greatly improved. When the user is a cold-start user, there are usually few social relationships, and the social relationships improve the performance of the recommendation algorithm. When the target item is a noncold-start item, the relationships between items can improve the accuracy of the recommendation algorithm. For the overall user set and cold-start user set, compared with PMF, the three social recommendation algorithms SoReg, TrustSVD, and TCRec show improved recommendation accuracies, indicating that the social relationship helps to improve the prediction accuracy of the recommender system, Compared with SoReg, TrustSVD and TCRec, DRMF performs better, indicating that combining

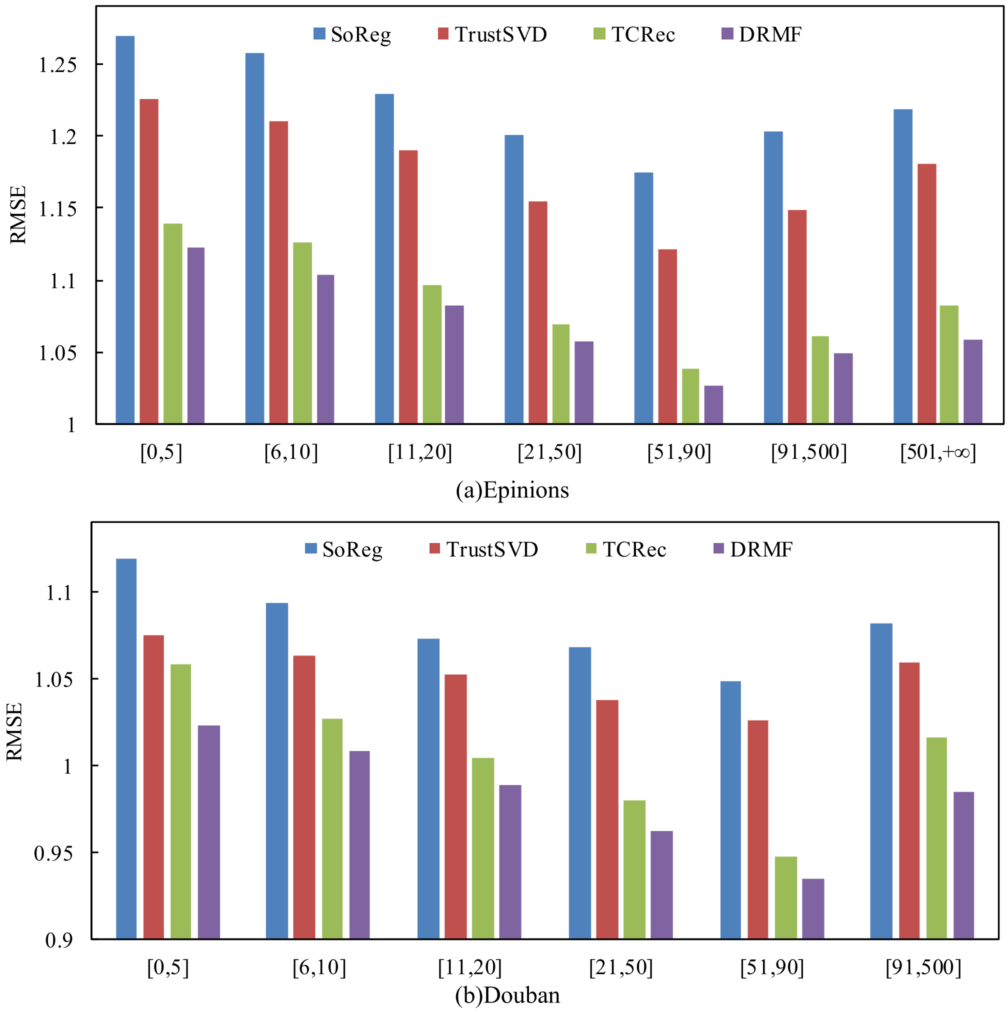


FIGURE 3. **Comparison of RMSE for four algorithms.**

item relationships and social relationships can improve the accuracy of recommendation.

Second, we conducted experiments and compared ratings. Taking the Epinions and Douban datasets as examples, the number of users in the training set was divided into five groups, i.e., [0, 10], [11, 30], [31, 50], [51, 90], and [91, $+\infty$], and the latent feature dimension was set to $d = 10$. Table 4 and Figure 2 show the proportion of each group of user ratings and the comparison results of the Epinions and Douban datasets, respectively. It can be seen from Table 4 that the majority of users have fewer than 10 ratings, indicating that the dataset is sparse. From the comparison results of the Douban and Epinions datasets, the SoReg, TrustSVD, TCRec, and DRMF algorithms perform better than the PMF algorithm, indicating that social relationships help to improve algorithm accuracy. DRMF is obviously superior to several recommended algorithms, indicating that, when the rating is sparse, the social relationships between users are usually sparse, and then considering the relationships between items can improve the accuracy of the recommendation algorithm. As the number of ratings increases, the RMSE trend of DRMF is not blindly reduced. Thus, when the number of ratings is too large, the user's preference will diverge, which results in the inability to accurately learn the latent feature vectors of user preferences. However, the DRMF algorithm in this paper still performs better than other algorithms.

Finally, we conducted experiments and compared different social relationships. Taking the Epinions and Douban datasets as examples, the number of social relationships in the training set is divided into five groups, i.e., [0, 5], [6, 10], [11, 20], [21, 50], [51, 90], [91, 500], and [501, $+\infty$], and the latent feature dimension is set to $d = 10$. Table 5 and Figure 3 show the distribution of social relations and the comparison results for the experimental datasets. This experiment compares the social recommendation methods SoReg, TrustSVD, TCRec and DRMF. It can be seen from Table 5 that most of the user social relationships are less than five. The comparison results on the Epinions and Douban datasets indicate that, when there are few social relationships, relying only on the rating data cannot result in good recommendations and will affect the recommendation accuracy. When there are many social relationships, the user's preferences will be overly dependent on the surrounding information, and the latent feature vectors of the learned users will be inaccurate. However, the DRMF method is superior to other algorithms, indicating that, when considering social relationships and items simultaneously, the proposed recommendation algorithm will achieve better results.

V. CONCLUSION

Recommendation algorithms which is based on social networks generally assume that the user's preference will be affected by the preferences of friends; however, in reality, social relationships are very sparse and unstable, and recommendations will be affected to varying degrees. To solve the problems of traditional social network recommendation algorithms, this paper proposes a double regularization matrix factorization recommendation algorithm. In addition to social relations, the relationships between items are considered, manifold learning is used to improve the item similarity calculation, and item regularization is used in the social matrix factorization model. Through experiments, this method has been shown to mitigate the cold start problem and the data sparseness problem.

This paper considers only the influence of attribute information on the relationships between items. In the future, a method that better integrates context information, item relationships and social relationships should be developed to better predict the target user's preferences for specific items. Finally, deep learning technology has shown great potential in the field of natural language processing. Thus, using deep learning techniques to improve existing recommendation algorithms will become a hot research direction in the field of recommender systems.

REFERENCES

[1] R. P. Karumur, T. T. Nguyen, and J. A. Konstan, "Personality, user preferences and behavior in recommender systems," *Inf. Syst. Frontiers*, vol. 20, no. 6, pp. 1241–1265, Dec. 2018.

[2] R. Salakhutdinovand and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, Jan. 2008, pp. 1257–1264.

[3] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 426–434.

[4] S. Biswas, L. V. S. Lakshmanan, and S. B. Ray, "Combating the cold start user problem in model based collaborative filtering," in *Proc. ACM Conf.*, Washington, DC, USA, Feb. 2017, pp. 1–11.

[5] H. F. Liu, L. P. Jing, and J. Yu, "Survey of matrix factorization based recommendation methods by integrating social information," *J. Softw.*, vol. 29, no. 2, pp. 340–362, Feb. 2018.

[6] B. Pal and M. Jenamani, "Trust inference using implicit influence for item recommendation," in *Proc. 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Dec. 2018, pp. 21–28.

[7] Q. Wu, M. Huang, and Y. Mu, "A collaborative filtering algorithm based on user similarity and trust," in *Proc. 14th Web Inf. Syst. Appl. Conf. (WISA)*, Nov. 2018, pp. 263–266.

[8] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, Hong Kong, Feb. 2011, pp. 287–296.

[9] G. B. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 29th AAAI Conf. Artif. Intell.*, Jan. 2015, pp. 123–129.

[10] J. Lee, G. Noh, H. Oh, and C.-K. Kim, "Trustor clustering with an improved recommender system based on social relationships," *Inform. Syst.*, vol. 77, pp. 118–128, Sep. 2018.

[11] X. Han, L. Wang, R. Farahbakhsh, Á. Cuevas, R. Cuevas, N. Crespi, and L. He, "CSD: A multi-user similarity metric for community recommendation in online social networks," *Expert Syst. Appl.*, vol. 53, pp. 14–26, Jul. 2016.

[12] Q. Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 199–210.

[13] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item cooccurrence," in *Proc. 10th Int. Conf. Recommender Syst.*, Sep. 2016, pp. 59–66.

[14] C. Xu, "A novel recommendation method based on social network using matrix factorization technique," *Inf. Process. Manage.*, vol. 54, no. 3, pp. 463–474, May 2018.

[15] L. Xiao, Z. Min, Z. Yongfeng, L. Yiqun, and M. Shaoping, "Learning and transferring social and item visibilities for personalized recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 337–346.

[16] M. Meng and J. Yu, "Zero-shot learning via robust latent representation and manifold regularization," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1824–1836, Apr. 2019.

[17] S. Gao, Z. Yu, T. Jin, and M. Yin, "Multi-view low-rank matrix factorization using multiple manifold regularization," *Neurocomputing*, vol. 335, pp. 143–152, Mar. 2019.

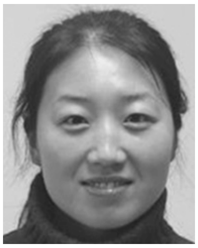
[18] B. Lecouat, C.-S. Foo, H. Zenati, and V. R. Chandrasekhar, "Semi-supervised learning with GANs: Revisiting manifold regularization," Jul. 2018, *arXiv:1805.08957*. [Online]. Available: https://arxiv.org/abs/1805.08957#

[19] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1403–1416, Jul. 2015.

[20] Y. Li, D. Wang, H. He, L. Jiao, and Y. Xue, "Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems," *Neurocomputing*, vol. 249, pp. 48–63, Aug. 2017.

RUIZHONG DU was born in 1975. He received the Ph.D. degree in engineering from Wuhan University. He is currently a Professor. His main research interests include network security, trusted computing, and network technology.

JIAOJIAO LU was born in 1993. Her main research interests include social networks and recommender systems.



HONGYUN CAI was born in 1980. She is currently pursuing the Ph.D. degree. She is also an Associate Professor. Her main research interests include information security, recommender systems, and user behavior analysis.