 CMC, vol.58, no.1, pp.101-120, 2019

An Entity-Association-Based Matrix Factorization Recommendation Algorithm

Gongshen Liu[1], Kui Meng[1,*], Jiachen Ding[1], Jan P. Nees[1], Hongyi Guo[1] and Xuewen Zhang[1]

Abstract: Collaborative filtering is the most popular approach when building recommender systems, but the large scale and sparse data of the user-item matrix seriously affect the recommendation results. Recent research shows the user's social relations information can improve the quality of recommendation. However, most of the current social recommendation algorithms only consider the user's direct social relations, while ignoring potential users' interest preference and group clustering information. Moreover, project attribute is also important in item rating. We propose a recommendation algorithm which using matrix factorization technology to fuse user information and project information together. We first detect the community structure using overlapping community discovery algorithm, and mine the clustering information of user interest preference by a fuzzy clustering algorithm based on the project category information. On the other hand, we use project-category attribution matrix and user-project score matrix to get project comprehensive similarity and compute project feature matrix based on Entity Relation Decomposition. Fusing the user clustering information and project information together, we get Entity-Association-based Matrix Factorization (EAMF) model which can be used to predict user ratings. The proposed algorithm is compared with other algorithms on the Yelp dataset. Experimental studies show that the proposed algorithm leads to a substantial increase in recommendation accuracy on Yelp data set.

Keywords: Collaborative filtering, matrix factorization, recommender system.

1 Introduction

With the rapid development of information technology and Internet, we gradually move into the era of big data. To extract valuable information from massive amount of data is a big challenge for both information consumer and information provider. The recommender system is developed to help users find their interested information, and let information provider target their customers more efficiently, which is a win-win solution. At present, the recommender system has been widely adopted to several Internet fields [Ricci, Rokach, Shapira et al. (2011)].

The recommendation algorithm is the core of a recommender system. The collaborative filtering (CF) recommendation algorithm is one of the most successful recommendation

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China.

[*] Corresponding Author: Meng Kui. Email: mengkui@sjtu.edu.cn.

technologies [Su and Khoshgoftaar (2009)]. The fundamental assumption of CF is that if user X and Y give similar rates to *n* items, or have similar behaviors (e.g. buying, watching, and listening), they will give similar rate on other items. CF recommendation algorithm divides users with similar behavior, and recommends items to them. Versus the content-based recommendation algorithm, CF recommendation algorithm is not limited by the content analysis technology. However, CF recommendation algorithm faces the challenges such as data sparsity, scalability, synonymy, and gray sheep [Su and Khoshgoftaar (2009)]. The rapid expansion of Internet and its users even make these worse.

Social network analysis shows that users of the same community often show a similar interest and behavior characteristics because of social factors [Krebs (2017)]. Therefore, in recent years the socialization recommender system with user's social attributes has become the research hotspot in recommender system. Usually traditional trust-based socialization recommendation algorithm only utilizes direct trust relationship among the users. With Internet scale increasing, the direct trust relationship between users becomes sparse inevitably. Moreover, the basic assumption of a trust-based socialized recommendation algorithm is that the user's interest preferences are similar to or are affected by their trusted users [Yang, Guo, Liu et al. (2014)]. In fact, the user's interest preferences are multifaceted, and usually are different from each other. A single direct social relationship could not characterize the difference exactly.

In this paper, we propose a recommendation algorithm named Entity-Association-based Matrix Factorization (EAMF) that integrates user information and project information together. We use overlapping community discovery algorithm to find user's community information, which avoids the sparseness of data caused by the use of direct social relations, and cluster user interest preference based on the project category information considering the user preference differences of the same community. These two kinds of information are merged together to cluster user character. On the other hand, we use project-category attribution matrix and user-project score matrix to get project comprehensive similarity and compute project feature matrix. Fusing the user clustered character and project feature matrix together, we get Entity-Association-based Matrix Factorization (EAMF) model which can be used to predict user ratings.

2 Related works

The traditional collaborative filtering recommendation algorithm is divided into memory-based method and model-based method [Bobadilla, Ortega, Hernando et al. (2013)]. In recent years the collaborative filtering recommendation algorithm based on the matrix decomposition model has been widely used as a branch of the model-based method. It can transform high-dimensional user-project scoring matrix into low-dimensional matrix product which represents the implicit eigenvector of user and project, and relieves accuracy decreasing caused by sparse data. The application of matrix decomposition technique in recommender system is first proposed in Koren et al. [Koren, Bell and Volinsky (2009)]. The probability matrix decomposition model is proposed in Salakhutdinov et al. [Salakhutdinov and Mnih (2007)].

In socialization recommendation algorithm, the matrix decomposition technique is combined with various social attribute of users. It gets better user implicit eigenvectors

by adding the correlation constraints in the process of matrix optimization decomposition, which assumes that user and their trusted users have similar interest preference or are affected by each other. SoRec model is proposed in Ma et al. [Ma, Yang, Lyu et al. (2008)]. It is a type of social spectrum regularization deformation method, which applies matrix decomposition technique to trust matrix. The user implicit eigenvector is generated during the decomposition optimization of user-project scoring matrix and trust matrix. STE model is proposed in Ma et al. [Ma, King and Lyu (2009)]. In STE, the items in the user-project scoring matrix are the combination of user's personal preference and their trusted friend's preferences. In the decomposition optimization, it get a weighted average of the users' score and his friends' score to the same item, which makes the recommendation results interpretable. SocialMF model is proposed in Jamali et al. [Jamali and Ester (2010)]. It assumes that the implicit eigenvector of the user is determined by the implicit eigenvector of his friend, and defines the concept of trust propagation during the decomposition optimization. However, these socialized recommendation algorithms use only the user's direct social relationship. When direct social relationship is sparse, the recommendation results will be unacceptable. Yang et al. improve SocialMF model [Yang, Steck and Liu (2012)]. They distinguish user category-specific social trust to different items and different friends, and divide friend circle according to the items they rated, which further intensify the problem of data sparsity. Guo et al. take the characteristics of trust diversity into consideration [Guo, Ma and Chen (2013)], and propose a trust strength aware social recommendation method, StrengthMF, assuming that a trust relation does not necessarily guarantee the similarity in preferences between two users. StrengthMF can acquire a better understanding of the relationship between trust relation and rating similarity, but the algorithm is still limited by the problem of direct social relations sparsity. Li et al. [Li, Wu, Tang et al. (2015)] introduce overlapping community discovery algorithm into the socialization recommendation algorithm. They focus on the constraint of regular term in objective function, and propose two models to reduce the impact of preference difference among users in the same community. Huang et al. propose an overlapping community detection algorithm named LEPSO [Huang, Li, Zhang et al. (2017)], which is a meta-heuristic approach, combining line graph theory, ensemble learning, and particle swarm optimization (PSO) together. They transform the overlapping community detection problem into a disjoint community detection problem on line graph, and use ensemble clustering techniques to optimize modularity of the line graph. Li et al. [Li, Zhang and Li (2017)] use both user generated contents and relationships between users to build a probabilistic user interests model, design a user interest propagation algorithm (UIP), and combine the UIP algorithm with classical matrix factorization to form a new rating prediction method, namely MF-UIP. Experimental studies show that MF-UIP outperforms existing algorithms.

3 Matrix factorization recommendation algorithm based on internal entity relationship

The flow of our Entity-Association-based Matrix Factorization (EAMF) recommendation algorithm is shown in Fig. 1.

Firstly, the overlapping community discovery algorithm is used to obtain the community structure of the social network, and so we get the user cluster based on community structure. According to project category information and user behavior record, the user

preference vector is synthesized with the category distribution vector and the category professional vector. Using the fuzzy C-mean clustering algorithm, we get the user cluster based on interest preferences. Then, we can quantify target user's preferences to above user clusters it belongs to, here is community structure based user cluster and interest preference based user cluster, and regularize these two user clusters respectively.

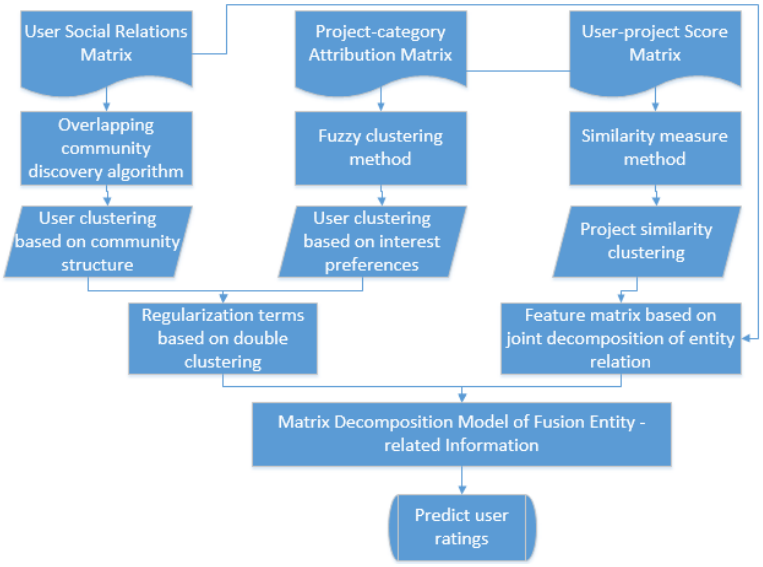


Figure 1: The flow chart of matrix factorization recommendation algorithm

On the other side, we calculate Scoring-based project similarity with users' historical rating information, and Category-based project similarity with project category information respectively. It is easy to obtain project relevance based on comprehensive similarity (project similarity clustering) and project feature matrix.

With these two kinds of information, user-based and project-based, we can use matrix decomposition model of fusion entity-related information to forecast the user's interest to the project.

3.1 Problem definition

We represent, $U = \{u_1, u_2, \ldots, u_m\}$ as the set of all users in the recommender system; $V = \{v_1, v_2, \ldots, v_n\}$ represent the set of all projects; $C = \{c_1, c_2, \ldots, c_q\}$ represent the set of all project categories; and m, n, q represents the number of users, projects and categories respectively. $R = (R_{ij})^{m \times n}, R_{ij} \in \{1, 2, 3, 4, 5\}$, represent user-project score matrix, and R_{ij} represents user u_i's score to the project v_j. $T = (T_{ij})^{m \times m}, T_{ij} \in \{0, 1\}$ represent the user's social relation matrix, and $T_{ij} = 1$ represents user u_i and user u_j are friends. We require a bidirectional confirmation of the user's social relations, so the matrix $\boldsymbol{T}$ is a symmetric matrix.

3.2 Clustering based on community structure

In recommender system, user clusters and the social links between users constitute a large social network. Usually, it is assumed that the user and its direct friends often have similar interests, or may influence each other. Base on such assumption, some researches [Ma, Yang, Lyu et al. (2008); Ma, King and Lyu (2009); Massa and Avesani (2004)] add user's social relations information to optimize traditional collaborative recommendation algorithms. However, in large-scale social network, there exists a long tail effect [Fortunato (2010)], that is, only a few social users have many social relations, and the vast majority of users only have a small number of social relations. Therefore, it is necessary to dig out other valuable information from the social network, where there is community structure in social network. Users in the same community share the same characteristics, such as similar geographical location, same industry sector, or common interest topics, and have more or less impact on each other. Moreover, users inevitably belong to multiple communities. Such overlapping communal information reflects different characteristics of a user.

Research on overlapping communities in social network is a hotspot in the field of community discovery in recent years [Wang, Liu, Pan et al. (2016); Wang, Liu, Li et al. (2017)]. Here we directly use the overlapping community discovery algorithm to obtain the social network in the recommendation system. BIGCLAM algorithm is an overlapping community discovery algorithm for large communal network [Yang and Leskovec (2013)]. It is based on the assumption that overlapping nodes are closely connected, and it is improved from the nonnegative matrix decomposition model. The experiment in Li et al. [Li, Wu, Tang et al. (2015)] show that using the community discovering result of BIGCLAM algorithm as constraint, the recommender system can obtain better recommendation results. Here we choose BIGCLAM algorithm to discover overlapping community in the user's social network.

User's interests among different communities are not the same. In Li et al. [Li, Wu, Tang et al. (2015)] they use average of all users' score vector in user-scoring matrix as community score vector. The similarity of the user's rating vector corresponding to the community score vector of the user in the community is represented as the degree of user's interest to the community. However, the contribution of each user to the community is different. Compared to users at the edge of the community structure, users who have more friends and direct social relationships are more representative to the community. Based on this hypothesis, we use the score vector of all users and the number of community friends to obtain the weighted community score vector.

$$Com(i) = \frac{\sum_{g \in \Omega(i)} |friend_i(g)| U_g^R}{\sum_{g \in \Omega(i)} |friend_i(g)|} \tag{1}$$

Here, $\Omega(i)$ represents all users in community i; $friend_i(g)$ represents user u_i's direct social friends in community i; U_g^R represents user score vector. According to Eq. (1), users who have more direct social relationships contribute more to the community score vector. Then, we calculate the Pearson correlation coefficient of the community score vector and the user score vector to obtain the similarity between them.

$$Sim(i,j)=\frac{\sum_{f\in A_{ij}}(U_{if}^{R}-\bar{U}_{i}^{R})\cdot(U_{jf}^{R}-\bar{U}_{j}^{R})}{\sqrt{\sum_{f\in A_{ij}}(U_{if}^{R}-\bar{U}_{i}^{R})^{2}}\cdot\sqrt{\sum_{f\in A_{ij}}(U_{jf}^{R}-\bar{U}_{j}^{R})^{2}}} \quad (2)$$

Where A_{ij} is non-zero collections of user score vector U_i^R and U_j^R. The output range of Pearson correlation coefficients are [-1, 1]. Here we use function *f(x)=(x+1)/2* to map the output range to [0, 1].

Thus, we get the user community information based on the social network structure. The similarity between the user score vector and the community-rating vector of the community represents the user's interest to the community.

3.3 Clustering based on interest preference

The overlapping community discovery algorithm divide users according to social network structure, and usually users belonging to the same community have similar characteristics or mutual influence. However, users in the same community may still have different preferences. For example, although those who love science fiction movies are grouped into one community, they may have different favor in music, game, travelling and so on. It is necessary to sub-group users in the same social network community. So, a fuzzy clustering algorithm based on interest preference is proposed. The algorithm utilizes the user's behavior record and the category of the project to find users who have similar preferences with the target user at generalization level.

3.3.1 Category distribution vector

The items that a user has rated may belong to different categories, and the user's ratings percentage of the items in a category is proportional to one's interest in that category. The distribution vector of the categories of all items that the user u_i has rated can be described as follows:

$$Dis(i)=(\frac{|P_i^{c_1}|}{|P_i|},\frac{|P_i^{c_2}|}{|P_i|},\ldots,\frac{|P_i^{cq}|}{|P_i|}) \quad (3)$$

Here, P_i is the item collection that the user u_i has rated; $P_i^{c_k}$ is the rated item collection by user u_i, which belongs to category c_k.

3.3.2 Category professional vector

In addition to the difference in the number of ratings for different categories, the more interest a user has in a category, the higher his or hers rating will be in this category. The well-known search engine algorithm, HITS, is used to calculate the user's professionality in a category.

User's historical behavior data is used to calculate one's professionality. However, one always has different professionality in different categories. It is necessary to discuss one's professionality respectively.

Here are two assumptions. First, if a user has rated a number of representative projects in a category, the user is familiar and professional with the category. Second, if a project has been rated by a lot of professional users, the project is representative in this category. So, for each category, one's professionality and project's representativity can mutually

reinforce each other.

For each category, one project's representativity is the professionality sum of the users who has rated the project, and the user's professionality is the representativity sum of projects that one has rated.

$$a_j^k = \sum_{u_i \in U} I_{ij} \times h_i^k$$

$$h_i^k = \sum_{v_j \in V^{c_k}} I_{ij} \times a_j^k \quad (4)$$

Here, a_j^k is the representation of project v_j belonging to category c_k; h_i^k is the professionality of user u_i to category c_k; I_{ij} is instruction function, if user u_i has rated project v_j, $I_{ij} = 1$, else $I_{ij} = 0$.

If User-project rating matrix is $R \in R^{m \times n}$, for category c_k, user-project rating matrix is $R_k \in R^{m \times n_k}$, where n_k is the number of projects belonging to category c_k; $a_k = (a_1^k, a_2^k, \dots, a_{n_k}^k)^T$ is a representation vector of the representativity of each project belonging to category c_k; $h_k = (h_1^k, h_2^k, \dots, h_m^k)^T$ is a professional vector representing the professionality of each user to category c_k.

For each category:

1) Initialize user professionality vector h_k and project representatively vector a_k, and make $\sum_{i=1}^{n_k} a_{k,i}^2 = 1,\ \sum_{i=1}^{m} h_{k,i}^2 = 1$

2) From Eq. (4), we can get a_k and h_k:

$$a_k = R_k^T \cdot h_k$$

$$h_k = R_k \cdot a_k \quad (5)$$

Assuming $a_k^{(t)}$、$h_k^{(t)}$ is a_k、h_k in the t_{th} round, then:

$$a_k^{(t)} = R_k^T \cdot R_k \cdot a_k^{(t-1)}$$

$$h_k^{(t)} = R_k \cdot R_k^T \cdot h_k^{(t-1)} \quad (6)$$

1) Normalize a_k and h_k, to make $\sum_{i=1}^{n_k} a_{k,i}^2 = 1,\ \sum_{i=1}^{m} h_{k,i}^2 = 1$

2) If $\left\| a_k^{(t+1)} - a_k^{(t)} \right\| < \varepsilon$ and $\left\| h_k^{(t+1)} - h_k^{(t)} \right\| < \varepsilon$ or $t > t_{max}$, then terminate, else go back to (2).

So, we get the professionality vector h_k for all users under each category. Choosing the value of specific position from each vector to form a vector, and normalizing it, we can get category professionality vector of respective specific position user.

$$Exp(i) = \left(\frac{h_i^1}{S_i}, \frac{h_i^2}{S_i}, \dots, \frac{h_i^{c_q}}{S_i}\right),\ S_i = \sum_{j=1}^{c_q} h_i^j \quad (7)$$

3.3.3 Category preference vector

The user's category preference has a positive relationship with user's rating distribution and professionality in that category. So, we can obtain the user category preference vector through weighting operation with the same position value of category distribution

vector and category professional vector.

$$Pre(i)_k = Dis(i)_k \times Exp(i)_k, k = 1, \dots, c_q \quad (8)$$

Here $Pre(i)_k$ is user u_i's preference to category c_k. The category preference vector of all users is used as the sample data in following clustering algorithm.

3.3.4 Clustering objective function

Fuzzy C-mean clustering [Bezdek, Ehrlich and Full (1984)] is a relatively mature algorithm in fuzzy clustering analysis. It optimizes the objective function to obtain the membership grade of each sample point to a cluster, that is, the sample point can belong to multiple clusters at the same time, according to the nature of user preference in the recommendation system. Minkowski distance is always used when calculating the similarity between the sample point vector and the cluster-like center point vector in the fuzzy C-mean clustering algorithm.

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}, X = (x_1, x_2, \dots, x_n) \in \mathrm{R},\ Y = (y_1, y_2, \dots, y_n) \in R \quad (9)$$

When $p = 1$, $D(X,Y)$ is Manhattan distance. When $p = 2$, it is Euclidean distance.

However, the user-project scoring matrix in the recommender system is sparse. Most users only score a few categories of projects, so the user category preference vector is sparse too. If we directly use Minkowski distance to calculate similarity, it might not get satisfying clustering result.

So we optimize similarity calculating method as follows:

$$D_{sparse}\left(Pre(i), g_j\right) = \frac{1}{\delta_i} \sum_{k=1}^{q} \delta_{i,k} (Pre(i)_k - g_{j,k}) \quad (10)$$

$$\delta_{i,k} = \begin{cases} 1, & if\ \ Pre(i)_k \neq 0 \\ 0, & if\ \ Pre(i)_k = 0 \end{cases}, \quad \delta_i = \sum_{k=1}^{q} \delta_{i,k}$$

We assume the number of clusters is l. In Eq. (10), g_j is the vector of a certain cluster $j (j = 1,2, \dots l)$. $g_{j,k}$ is the kth element in g_j. We calculate the similarity from those non-zero elements in user category preference vector.

The objective function is defined as bellow.

$$Obj(B,G) = \sum_{i=1}^{m} \sum_{j=1}^{l} u_{ij}^{\theta} D_{sparse}\left(Pre(i), g_j\right) \quad (11)$$

Here, $u_{ij} \in [0,1]$ is user u_i 's membership grade to cluster ψ_j , and u_{ij} satisfies $\sum_{j=1}^{l} u_{ij} = 1$. $B = (u_{ij})^{m \times l}$, which is user cluster membership matrix. $G = (g_1, g_2, \dots, g_l)^T$ is cluster center matrix, and $\theta \in [0, \infty]$ is Clustering fuzzy degree, usually it is set as 2 [Xu and Wunsch (2005)].

3.3.5 Clustering algorithm

The preference-based fuzzy clustering algorithm updates the user cluster membership matrix ***B*** and the cluster center matrix ***G*** by iteration, and approaches the objective function until the error value of the objective function converge to the preset threshold.

1) Randomly initialize the user cluster membership matrix U to satisfy $\sum_{j=1}^{l} u_{ij} = 1$, choose cluster amount l and clustering ambiguity θ, and determine the convergence threshold $\varepsilon \in (0,1)$ and maximum number of iterations t_{max}.

2) Update the cluster center matrix $\boldsymbol{G}$

$$g_j = \sum_{i=1}^{m} u_{ij}^{\theta} Pre(i) / \sum_{i=1}^{m} u_{ij}^{\theta}, j=1,2,\ldots,l \quad (12)$$

3) Update the cluster membership matrix $\boldsymbol{B}$

$$u_{ij} = 1 \Big/ \sum_{k=1}^{l} \left(\frac{D_{sparse}(Pre(i), g_j)}{D_{sparse}(Pre(i), g_k)} \right)^{\frac{1}{\theta-1}},$$

$i=1,2,\ldots,m; j=1,2,\ldots,l$ (13)

4) If $\left\| B^{(t+1)} - B^{t} \right\| < \varepsilon$ or t > t_{max}, then terminates the iteration, else back to (2).

The element u_{ij} in user cluster membership matrix $\boldsymbol{B}$ represents user u_i's interest degree to interest cluster ψ_j. Users in the same cluster have similar general preferences.

3.4 Project relevance based on comprehensive similarity

In traditional project-based collaborative filtering recommendation algorithm, the project similarity is obtained by calculating the rating similarity among common user sets in the user-project scoring matrix. However, such method requires that there are plenty of users, who have scored two items at the same time. Otherwise, we will not obtain accurate project similarity. Moreover, this approach does not take the properties of the project itself into account. Project category information is important to describe the basic information of the project. Compared with other projects, the project belonging to the same category are obviously more similar to each other. The reliability of the algorithm can be improved effectively if project category information is considered in the similarity calculation method.

3.4.1 Scoring-based project similarity

First, we obtain project similarity based on scoring by the column vector in the user-project scoring matrix. Elements in the column vector represent users' score to that project. The scoring-based project similarity is calculated by improved Pearson correlation coefficient. We add user activity constraint in Pearson correlation coefficient calculating.

$$Rsim(i,j) = \frac{\sum_{u \in \Lambda_{ij}} (R_{ui} - \overline{R}_i) \cdot (R_{uj} - \overline{R}_j) \cdot (\log(1 + |N(u)|))^{-1}}{\sqrt{\sum_{u \in \Lambda_{ij}} (R_{ui} - \overline{R}_i)^2} \cdot \sqrt{\sum_{u \in \Lambda_{ij}} (R_{uj} - \overline{R}_j)^2}} \quad (14)$$

Here, Λ_{ij} represents the position set where all elements in both project rating R_i and R_j are not zero, *u* represents the user who has rated project *i* and *j*, *N(u)* represents the rated project set by user *u*. We can use *f(x)=(x+1)/2* to map the output to [0,1]. From Eq. (14), it is obvious that the higher the user activity is, the less the user can contribute to the project similarity.

3.4.2 Category-based project similarity

Usually many projects in the recommender system belong to multiple categories. The project can be presented as a vector of Boolean values on the category dimension, and each Boolean value reflects the project category information. We assumed that the more projects are included in one category, the greater generality the category is, and the less similarity the included projects have. The category-based project similarity is calculated using improved cosine similarity, which bases on the category generality. Thus, a category generality constraint is added to the cosine similarity.

$$Csim(i,j) = \frac{\sum_{c\in C} \delta_{ic}\cdot\delta_{jc}\cdot(1+\frac{|N(c)|}{|N(all)|})^{-1}}{\sqrt{\sum_{c\in C}\delta_{ic}^2}\cdot\sqrt{\sum_{c\in C}\delta_{jc}^2}} \quad (15)$$

Here, $\boldsymbol{C}$ represents category collection, and δ_{ic} is a Boolean, if project *j* belongs to category *c*, then $\delta_{ic} = 1$, else $\delta_{ic} = 0$. N(C) represent project collection in category *c*, and *N(all)* represents all project collection. Formula (15) shows that the greater generality a project is, the less it could contribute to the project category similarity.

We choose the larger one from scoring-based project similarity and category-based project similarity. After normalization, project relevance based on comprehensive similarity is obtained.

Sim(i,j) =max(Rsim(i,j), Csim(i,j)) (16)

Considering about computational complexity, only top 100 projects with the highest similarity to each project are recorded.

3.5 Matrix decomposition method based on entity relation decomposition

3.5.1 Probability matrix decomposition model

Matrix decomposition model is widely used in collaborative filtering recommendation algorithms. It decomposes the user-project scoring matrix $\boldsymbol{R}$ into two low-dimensional matrix products.

$$R \approx U^T V \quad (17)$$

Where $U \in \mathbb{R}^{k\times m}$, $V \in \mathbb{R}^{k\times n}$, $k \ll \min(m,n)$. The forecast score user u_i gives to project v_j is the transpose of the ith column in low-dimensional matrix $\boldsymbol{U}$, U_i^T, multiples by the j-th column in $\boldsymbol{V}$, V_j. So, U_i is called as user implicit feature vector and V_j as project implicit feature vector. The PMF model is also decomposing user-project score matrix $\boldsymbol{R}$ into the product of user implicit feature vector $\boldsymbol{U}$ and project implicit feature vector $\boldsymbol{V}$. User's number is *m* and project's number is *n*, which is $R \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{k\times m}$, $V \in \mathbb{R}^{k\times n}$, $k \ll \min(m,n)$. Assume that the conditional distribution probability of the observable data in the score matrix is as following:

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}\left[N(R_{ij}|U_i^T V_j,\sigma_R^2)\right]^{I_{ij}^R} \quad (18)$$

Here, $N(x|u,\sigma^2)$ represents the probability density function of the Gaussian distribution with mean *u* and variance σ^2. I_{ij}^R is an instruction function. If user u_i has scored project v_j, then I_{ij}^R=1, else I_{ij}^R=0.

The implicit feature matrix of both the user and the project satisfy the Gaussian transcendental with a mean of zero.

$$p(U|\sigma_U^2) = \prod_{i=1}^{m} N(U_i|0, \sigma_U^2 I)$$
$$p(V|\sigma_V^2) = \prod_{j=1}^{n} N(V_j|0, \sigma_V^2 I) \quad (19)$$

3.5.2 Matrix decomposition method based on entity relation decomposition

In fact, friends influence one's consumption choice and users are more willing to believe their friends recommendations. The relationship between projects also impact on user consumption decision. Users are usually more interested in the project that is similar to his favor ones. A matrix decomposition model based on entity relation decomposition is proposed. The model uses the user social matrix ***T*** and the project similarity matrix ***S***, and optimizes the implicit feature matrix of these two entities with joint decomposition of the scoring matrix ***R***.

The improved matrix decomposition model is shown in Fig. 2. The user social relations matrix ***T*** characterizes the information between users. Like SoRec model, we decompose T into user implicit feature matrix ***U*** and social relationship implicit feature matrix ***P***, where $P \in \mathbb{R}^{k\times m}$. In addition, the project similarity matrix ***S*** represents relationship between projects, and it can be decomposed into project implicit feature matrix ***V*** and similarity implicit feature matrix ***Q***, where $Q \in \mathbb{R}^{k\times n}$.

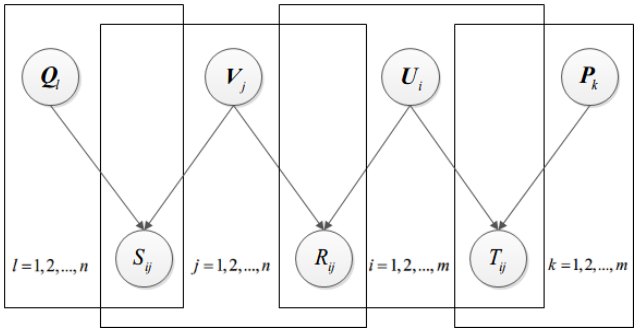


Figure 2: Entities relationship based matrix factorization model

Suppose the user social relations matrix ***T*** and the project similarity matrix ***S*** have the following conditional distribution probability:

$$p(T|U, P, \sigma_T^2) = \prod_{i=1}^{m}\prod_{k=1}^{m}\left[N(T_{ik}|U_i^{\mathrm{T}} P_k, \sigma_T^2)\right]^{I_{ik}^T}$$
$$p(S|V, Q, \sigma_S^2) = \prod_{j=1}^{n}\prod_{l=1}^{m}\left[N\left(TS_{jl}\middle|V_j^{\mathrm{T}} Q_l, \sigma_S^2\right)\right]^{I_{jl}^S} \quad (20)$$

Here, I_{ik}^T and I_{jl}^S are instruction functions. If user u_i trusts user u_k, then $I_{ik}^T = 1$, else $I_{ik}^T = 0$. If the corresponding position in the project comprehensive similarity matrix has a value, then $I_{jl}^S = 1$, else $I_{jl}^S = 0$.

Suppose that social relationship implicit feature matrix $\boldsymbol{P}$ and similarity implicit feature matrix $\boldsymbol{Q}$ also satisfy the Gaussian priori with the mean equals zero.

$$p(P|\sigma_P^2) = \prod_{i=1}^{m} N(P_i|0,\sigma_P^2 I)$$
$$p(Q|\sigma_Q^2) = \prod_{j=1}^{n} N(Q_j|0,\sigma_Q^2 I) \quad (21)$$

So, we can join user rating information, user social relations and project similarity together through the shared implicit feature space. Using Bayesian derivation, we can obtain joint posterior probability distribution of matri**x *U, V, P, Q.***

$$p(\mathrm{U},\mathrm{V},\mathrm{P},\mathrm{Q}|R,T,S,\sigma_R^2,\sigma_T^2,\sigma_S^2,\sigma_U^2,\sigma_V^2,\sigma_P^2,\sigma_Q^2) \propto$$
$$p(R|U,V,\sigma_R^2)p(T|U,P,\sigma_T^2)p(S|V,Q,\sigma_S^2)p(U|\sigma_U^2)p(V|\sigma_V^2)p(P|\sigma_P^2)p(Q|\sigma_Q^2) =$$
$$\prod_{i=1}^{m}\prod_{j=1}^{n}\left[N(R_{ij}|U_i^{\mathrm{T}}V_j,\sigma_R^2)\right]^{I_{ij}^R} \times \prod_{i=1}^{m}\prod_{k=1}^{m}\left[N(T_{ik}|U_i^{\mathrm{T}}P_k,\sigma_T^2)\right]^{I_{ij}^T} \times$$
$$\prod_{j=1}^{n}\prod_{l=1}^{n}\left[N(S_{jl}|V_j^{\mathrm{T}}Q_l,\sigma_S^2)\right]^{I_{jl}^S} \times \prod_{i=1}^{m} N(U_i|0,\sigma_U^2 I) \times \prod_{j=1}^{n} N(V_j|0,\sigma_V^2 I) \times$$
$$\prod_{i=1}^{m} N(P_i|0,\sigma_P^2 I) \times \prod_{j=1}^{n} N(Q_j|0,\sigma_Q^2 I) \quad (22)$$

In order to maximize the joint posterior probability distribution, we take logarithm of above equation.

$$\ln P\left(\mathrm{U},\mathrm{V},\mathrm{P},\mathrm{Q}|R,T,S,\sigma_R^2,\sigma_T^2,\sigma_S^2,\sigma_U^2,\sigma_V^2,\sigma_P^2,\sigma_Q^2\right)$$
$$= -\frac{1}{2\sigma_R^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\left(R_{ij} - U_i^{\mathrm{T}}V_j\right)^2$$
$$-\frac{1}{2\sigma_T^2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^T\left(T_{ik} - U_i^{\mathrm{T}}P_k\right)^2$$
$$-\frac{1}{2\sigma_S^2}\sum_{j=1}^{n}\sum_{l=1}^{n} I_{jl}^S\left(S_{jl} - V_j^{\mathrm{T}}Q_l\right)^2$$
$$-\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} U_i^{\mathrm{T}}U_i - \frac{1}{2\sigma_V^2}\sum_{i=1}^{n} V_i^{\mathrm{T}}V_i$$
$$-\frac{1}{2\sigma_P^2}\sum_{i=1}^{m} P_i^{\mathrm{T}}P_i - \frac{1}{2\sigma_Q^2}\sum_{i=1}^{n} Q_i^{\mathrm{T}}Q_i$$
$$-\frac{1}{2}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\right)\ln\sigma_R^2 - \frac{1}{2}\left(\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^T\right)\ln\sigma_T^2$$
$$-\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} I_{ij}^S\right)\ln\sigma_S^2$$
$$-\frac{1}{2}\left(md\ln\sigma_U^2 + nd\ln\sigma_V^2 + md\ln\sigma_P^2 + nd\ln\sigma_Q^2\right) + \boldsymbol{C} \quad (23)$$

Here, $\boldsymbol{C}$ is a constant that does not depend on the model parameters. To maximize posteriori probability of $\boldsymbol{U},\boldsymbol{V},\boldsymbol{P},\boldsymbol{Q}$ is equivalent to minimize the following error square and objective functions.

$$L = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R\left(R_{ij} - U_i^{\mathrm{T}}V_j\right)^2 + \frac{\lambda_T}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} I_{ij}^T\left(T_{ij} - U_i^{\mathrm{T}}P_j\right)^2 +$$
$$\frac{\lambda_S}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} I_{ij}^S\left(S_{ij} - V_i^{\mathrm{T}}Q_j\right)^2 + \frac{\lambda_U}{2}\|U\|_{Fro}^2 + \frac{\lambda_V}{2}\|V\|_{Fro}^2 + \frac{\lambda_P}{2}\|P\|_{Fro}^2 + \frac{\lambda_Q}{2}\|Q\|_{Fro}^2 \quad (24)$$

Where,$\lambda_T = \sigma_R^2/\sigma_T^2$, $\lambda_S = \sigma_R^2/\sigma_S^2$, $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_P = \sigma_R^2/\sigma_P^2$, $\lambda_Q = \sigma_R^2/\sigma_Q^2$.

So, a matrix decomposition model based on entity relation decomposition is obtained.

3.5.3 Regularization terms based on double clustering

In real life, the decisions we make are often influenced by friends or domain authorities. In Sections 3.2 and 3.3, we get the user social network community clustering information and user generalization interest preference clustering information. The former gathers together users who interact with each other and who have the same characteristics. The latter gathers together users who have similar interest preference in multi-domain. It is clear that the similarity of the target user with the user in the same set is higher than that of the user with who does not share any of the sets. Besides, the user's interest preference is close to the average interest preference of all the users in the same set. And the user interest degree in different sets is different. Based on the above assumptions, we improve the matrix decomposition model proposed in Ma et al. [Ma, Zhou, Liu et al. (2011)] and introduce new regular items.

$$\lambda_Z \sum_{i=1}^{m} \sum_{h=1}^{c_n} I_{ih}^N S_{ih} \sum_{g=1}^{c_p} I_{ig}^P Z_{ig} \left\| U_i - \frac{1}{|\Omega_{h,g}(i)|} \sum_{f \in \Omega_{h,g}(i)} U_f \right\|_{Fro}^2 \tag{25}$$

Here, $\lambda_Z (\lambda_Z > 0)$ is the coefficient of adjusting clustering regularization. I_{ih}^N is an instruction function. If user u_i is in the community $\Upsilon_h (h = 1,2, \ldots, c_n)$, then $I_{ih}^N = 1$, else $I_{ih}^N = 0$. S_{ih} represent user u_i's interest degree to community Υ_h. $I_{ig}^P$is also an instruction function. If user u_i is in interest preference cluster $\Psi_g (g = 1,2, \ldots, c_p)$, then $I_{ig}^P = 1$, else $I_{ig}^P = 0$. Z_{ig}represents user u_i's interest degree to interest preference cluster Ψ_g. $\Omega_{h,g}(i)$ represents the user collection who are in the same social networking community Υ_h and the same interest preference cluster Ψ_g as user u_i. At this point, a regular term based on user double clustering is obtained.

$$L = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R \left(R_{ij} - U_i^{\mathrm{T}} V_j \right)^2 + \frac{\lambda_U}{2} \|U\|_{Fro}^2 + \frac{\lambda_V}{2} \|V\|_{Fro}^2 +$$
$$\lambda_Z \sum_{i=1}^{m} \sum_{h=1}^{c_n} I_{ih}^N S_{ih} \sum_{g=1}^{c_p} I_{ig}^P Z_{ig} \left\| U_i - \frac{1}{|\Omega_{h,g}(i)|} \sum_{f \in \Omega_{h,g}(i)} U_f \right\|_{Fro}^2 \tag{26}$$

3.5.4 Entity-Association-based matrix factorization

We add the user double clustering regular term into the matrix decomposition model based on entity relation, and get the final matrix decomposition model, Entity-Association-based Matrix Factorization (EAMF). The objective function is shown in Eq. (27).

$$L = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R \left(R_{ij} - U_i^{\mathrm{T}} V_j \right)^2 + \frac{\lambda_T}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} I_{ij}^T \left(T_{ij} - U_i^{\mathrm{T}} P_j \right)^2 +$$
$$\frac{\lambda_S}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} I_{ij}^S \left(S_{ij} - V_i^{\mathrm{T}} Q_j \right)^2 + \frac{\lambda_Z}{2} \sum_{i=1}^{m} \sum_{h=1}^{c_n} I_{ih}^N S_{ih} \sum_{g=1}^{c_p} I_{ig}^P Z_{ig} \left\| U_i - \right.$$
$$\left. \frac{1}{|\Omega_{h,g}(i)|} \sum_{f \in \Omega_{h,g}(i)} U_f \right\|_{Fro}^2 + \frac{\lambda_U}{2} \|U\|_{Fro}^2 + \frac{\lambda_V}{2} \|V\|_{Fro}^2 + \frac{\lambda_P}{2} \|P\|_{Fro}^2 + \frac{\lambda_Q}{2} \|Q\|_{Fro}^2$$
(27)

The local optimal solution of user implicit feature matrix $\boldsymbol{U}$ and project implicit feature matrix $\boldsymbol{V}$ are obtained by stochastic gradient descent method. The corresponding partial

derivative is shown in Eq. (28).

$$\frac{\partial L}{\partial U_i} = \sum_{j=1}^{n} I_{ij}^{R} V_j (U_i^{\mathrm{T}} V_j - R_{ij}) + \lambda_P \sum_{k=1}^{m} I_{ik}^{T} P_k (U_i^{\mathrm{T}} P_k - T_{ik}) + \lambda_U U_i +$$
$$\lambda_Z \sum_{k=1}^{c_n} I_{ik}^{N} S_{ik} \sum_{g=1}^{c_p} I_{ig}^{P} Z_{ig} \left(U_i - \frac{1}{|\Omega_{k,g}(i)|} \sum_{f \in \Omega_{k,g}(i)} U_f\right) +$$
$$\lambda_Z \sum_{k=1}^{c_n} \sum_{g=1}^{c_p} \sum_{q \in \Omega_{k,g}(i)} \frac{S_{qk} Z_{qg}}{|\Omega_{k,g}(q)|} \left(\frac{1}{|\Omega_{k,g}(q)|} \sum_{f \in \Omega_{k,g}(q)} U_f - U_q\right)$$
$$\frac{\partial L}{\partial V_j} = \sum_{i=1}^{m} I_{ij}^{R} V_j (U_i^{\mathrm{T}} V_j - R_{ij}) + \lambda_Q \sum_{l=1}^{n} I_{jl}^{S} Q_l (V_j^{\mathrm{T}} Q_l - S_{jl}) + \lambda_V V_j$$
$$\frac{\partial L}{\partial P_k} = \lambda_P \sum_{i=1}^{m} I_{ik}^{T} U_i (U_i^{\mathrm{T}} P_k - T_{ik}) + \lambda_P P_k$$
$$\frac{\partial L}{\partial Q_l} = \lambda_Q \sum_{i=1}^{n} I_{il}^{T} V_i (V_i^{\mathrm{T}} Q_l - S_{il}) + \lambda_Q Q_l \quad (28)$$

The elements in $\boldsymbol{U}$ and $\boldsymbol{V}$ are updated along the gradient descending direction through iterations.

4 Experiments and evaluation

4.1 Data set

Yelp.com is one of the largest local business review sites in the world. It not only allows users to review or rate merchants, but also is an Internet company with distinguished social characteristics. It encourages active interaction between users, and a user can create a two-way confirmation of friendship with other users. The dataset we use to test the proposed algorithm is provided by Yelp. The dataset consists of 84,541 users, 43,252 items, 918,617 project scores and 725,603 two-way friend relationship information. Among them, the project score is an integer from 1 to 5, and all projects are divided into 12 categories. Detail information of the dataset is shown in Tab. 1.

4.2 Contrast algorithm

In order to verify the accuracy difference between the proposed algorithm and other algorithms, we choose the following algorithms as contrast algorithms.

BaseMF: Basic matrix decomposition model proposed in Bobadilla et al. [Bobadilla, Ortega, Hernando et al. (2013)], which does not add user social relationship information or project category information.

SocialMF: Matrix decomposition model combining user trust relationship information proposed in Jamali et al. [Jamali and Ester (2010)]. It assumes that the user vector is determined by his friends' user vector. And the concept of trust propagation is introduced into the decomposition optimization process.

SoReg: The concept of socialized regularity is first added into matrix decomposition model in Ma et al. [Ma, Zhou, Liu et al. (2011)]. So, the user's preference is similar to the average of his friends.

MFC: Overlapping community discovery algorithm is added into matrix decomposition model in Li et al. [Li, Wu, Tang et al. (2015)]. It distinguishes community difference on the basis of SoReg algorithm.

CircleCon: In Yang et al. [Yang, Steck and Liu (2012)], it is assumed that one's trust to

his friends is different according to different category. They divide the user trust network via project category on the basis of the SocialMF algorithm.

Table 1: Configuration of nodes

Category	User Count	Item Count	Rating Count	Sparsity
Active Life	19083	5244	65839	6.579E-04
Arts	9276	5945	49165	8.915E-04
Bars	11472	3177	42476	1.165E-03
Beauty	17685	4521	57628	7.208E-04
Education	1672	380	2187	3.422E-03
Food	27528	16139	325931	7.337E-04
Hotels	15441	1978	58514	1.946E-03
Night Life	23154	10633	189670	7.704E-04
Pets	6064	1527	8351	9.017E-04
Restaurants	22725	18326	369670	8.877E-04
Services	12443	1801	23991	1.071E-03
Shopping	14792	9274	32546	2.344E-04

4.3 Evaluation index

Here we use five cross validation method. The data set is randomly divided into five parts. In each experiment, four of them are selected as a training set and the remaining one is a test set. The final evaluation data is the average of 5 tests.

We use MAE (mean absolute error) and RMSE (root mean square error) as the evaluation criteria.

$$MAE = \frac{\sum_{(i,j)\in R_{test}} |R_{ij} - \hat{R}_{ij}|}{|R_{test}|} \tag{29}$$

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in R_{test}} (R_{ij} - \hat{R}_{ij})^2}{|R_{test}|}} \tag{30}$$

Here, R_{test} represent all users and projects in test set. R_{ij}Rij represent the real score user u_i gave to project v_j. $\hat{R}_{ij}$ represent the forecast score user u_i will give to project v_j. Rtest represent the number of scores in test set. The smaller ***MAE*** and ***RMSE*** are, the more accurate the recommendation is.

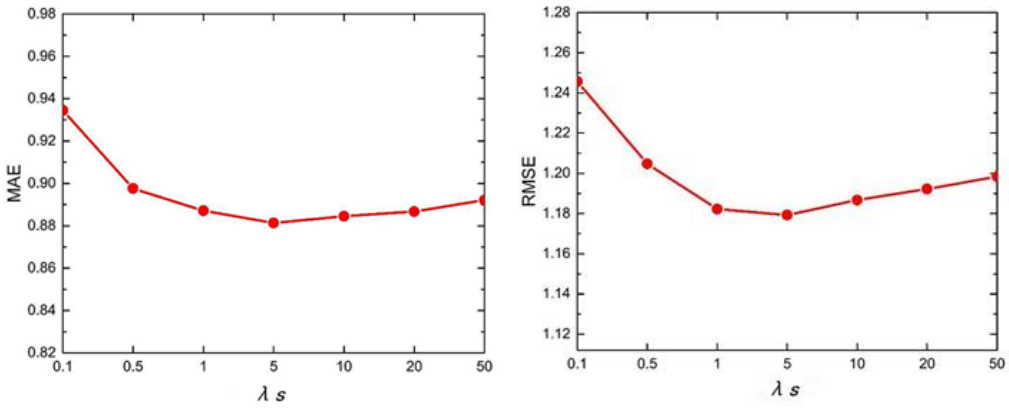
4.4 Experiment results and analysis

4.4.1 Determine the weighting factor λ_s

λ_s represent the proportion of project comprehensive similarity in matrix decomposition model, which is the recommended method dependence on project relevance. If λ_s equals 0, it means that the algorithm does not consider about project relevance, just like SoRec model. If λ_s approaches ∞, it means that only project relevance is considered. If λ_s equals other values, it means that it is considered in the algorithm about association

influence among user direct social relations, project similarity, and project score matrix. Here, we take λ_s as {0.1, 0.5, 1, 5, 10, 20, 50} and the results are shown in Fig. 3.

It can be seen in Fig. 3, that λ_s influents the recommendation accuracy greatly. If λ_s is small, MAE and RMSE are high, which means low accuracy. As λ_s increasing, MAE and RMSE gradually increase correspondingly. It indicates that project-related information plays a positive role. When λ_s is 5, the best result is obtained. After that the accuracy reduces again. The reason may be that excessive consideration of project relevance influence to results leads to the decrease of implicit eigenvector proportion in score matrix decomposition.



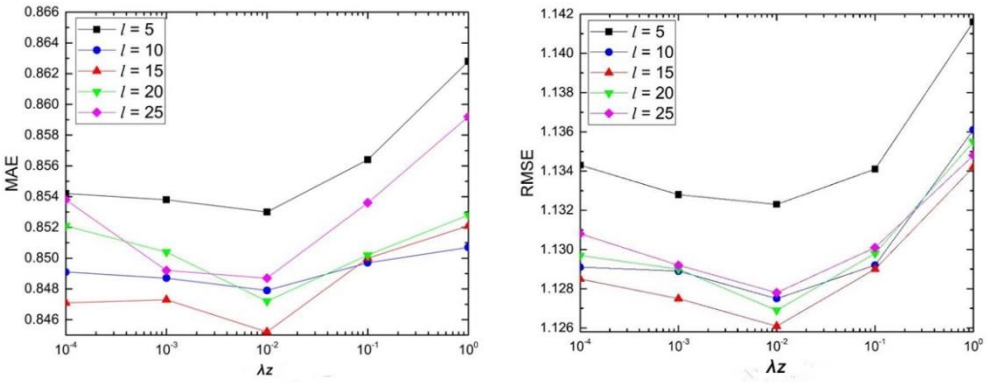
(a) *MAE* (b) *RMAE*

Figure 3: Impact of parameter $\boldsymbol{\lambda_s}$

4.4.2 Determine the number of interest cluster l

l represents the number of generalized interest clusters that are divided by all users' behavior records and project category information. Here, we choose *l* from 5 to 25 with Step 5 and record *MAE* and *RMSE* with different *l*. In order to understand the impact of *l* on the recommended results, we take five groups of experiments with different regular term coefficient λ_Z. The results are shown in Fig. 4.

It can be seen from Fig. 3, for different λ_Z, the accuracy is nearly the same with different *l*. If *l* is too large or too small, it will have a negative impact on the recommended results. When *l* is 15, MAE and RMSE are minimal at the same time. If *l* is too small, the fuzzy clustering results cannot clearly distinguish users at different levels of interest. Similarly, if *l* is too large, there will be too many clusters, which may weaken the expression of user's general interest.

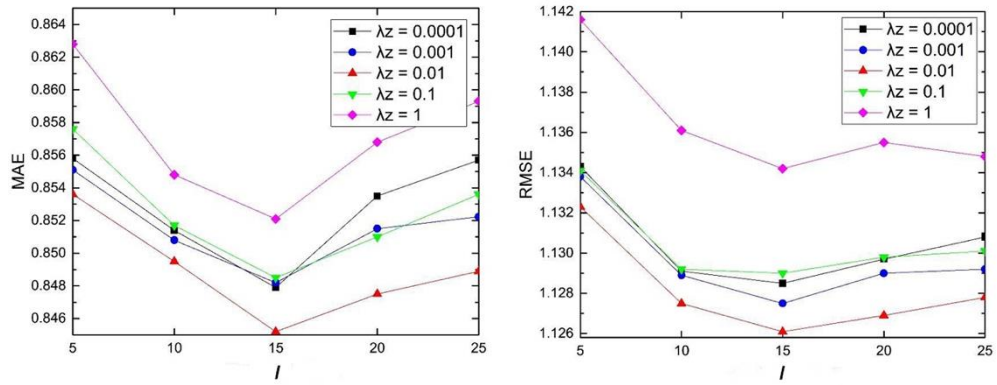


(a) MAE (b) RMSE

Figure 4: Impact of parameter *l*

4.4.3 Determine regular term coefficient λ_Z

λ_Z represent the proportion of user social network overlapping community information and the interest preference fuzzy clustering information in matrix decomposition model. When λ_Z is 0, the proposed model is equivalent to the basic matrix decomposition model. Take the value of λ_Z as $\{0.0001,0.001,0.01,0.1,1\}$, and record MAE and RMSE with different λ_Z. Similarly, we took five group of experiments with different *l* to show λ_Z's influence in recommendation. The result is shown in Fig. 5.



(a) MAE (b) RMSE

Figure 5: Impact of parameter λ_Z

As can be seen from Fig. 4, for different *l*, the accuracy is almost the same with different λ_Z. When λ_Z is small, MAE and RMSE are relatively high. As λ_Z increases, MAE and RMSE decrease. The optimal accuracy is achieved when λ_Z is 0.01. After that, the accuracy reduces again.

We think it is concerned with that additional information we used in the algorithm. When λ_Z is too small, the additional information has little effect to obtain better user implicit

eigenvector. Likewise, when λ_Z is too large, the additional information effect too great in vector optimization.

4.4.4 Result comparison of different recommended algorithms

We can see that when λ_s is 5, l is 15 and λ_Z is 0.01, our proposed algorithm, EAMF, can obtain the best recommendation result. Moreover, we compare EAMF algorithm with related algorithms described in Section 3.2. First, five cross validations determine the parameter of all algorithms in the experiment. Both regular term coefficients λ_U and λ_V are 0.01, and the dimension of user and project implicit eigenvector is 10. In SocialMF, SoReg, MFC, CircleCon algorithm, social regular coefficient λ_Z is set as 0.01, 0.01, 0.001, 0.01. β in SoReg algorithm is set as 0.5. The results are shown in Tab. 2.

Table 2: Comparisons of EAMF and other methods

Method	***MAE***	***RMSE***
BaseMF	1.2125	1.6168
SocialMF	1.0823	1.4334
SoReg	0.9901	1.2686
MFC	0.9474	1.1954
CircleCon	0.9563	1.2221
EAMF (this paper)	0.8452	1.1263

We can find from Tab. 2, EAMF algorithm, proposed in this paper, is more accurate than other algorithms. The result of BaseMF algorithm is worst because it only takes user-project scoring into account. SocialMF, SoReg, MFC algorithms do not simultaneously use user social information and project category information. The CircleCon algorithm directly divides the user by projects which are rated, while the user social relationship under a single project category may be sparse. EAMF algorithm model take project relevance as an influencing factor that is as important as user relevance, and optimizes the measurement of user relevance in the above-mentioned socialized recommendation algorithms. Therefore, it gets better recommended results.

5 Conclusion

Most traditional socialization recommendation algorithms just base on direct social relations which face with the problem of sparse social information. Because they do not take into account the impact of user interest preferences, those algorithms lead to high MAE and RMSE. Moreover, most algorithms only focus on user's characters while ignoring that project attribute is also important in item rating.

We propose an Entity-Association-based Matrix Factorization recommendation algorithm which fuses user information and project information together. It clusters users by social relations and interest preference respectively, characterizes projects by user's rating history and project category information, and combine the project feature matrix with user clustering matrix to predict user ratings. Experiment shows that considering the project feature, we can get better recommended results.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (61772337, 61472248 and U1736207), the SJTU-Shanghai Songheng Content Analysis Joint Lab, and program of Shanghai Technology Research Leader (Grant No. 16XD1424400).

References

Bezdek, J. C.; Ehrlich, R.; Full, W. (1984): The fuzzy c-means clustering algorithm. *Computers & Geosciences*, vol. 10, no. 2, pp. 191-203.

Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez A. (2013): Recommender systems survey. *Knowledge-Based Systems*, vol. 46, no. 1, pp. 109-132.

Fortunato, S. (2010): Community detection in graphs. *Physics Reports*, vol. 486, no. 3, pp. 75-174.

Guo, L.; Ma, J.; Chen, Z. (2013): Trust strength aware social recommendation method. *Journal of Computer Research & Development*, vol. 50, no. 9, pp. 1805-1813.

Huang, F.; Li, X.; Zhang, S. (2017): Overlapping community detection for multimedia social networks. *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1881-1893.

Jamali, M.; Ester, M. (2010): A matrix factorization technique with trust propagation for recommendation in social networks. *ACM Conference on Recommender Systems*, pp. 135-142.

Krebs, V. (2017): Social network analysis: an introduction. http://www.orgnet.com/sna.html.

Koren, Y.; Bell, R.; Volinsky, C. (2009): Matrix factorization techniques for recommender systems. *Computer*, vol. 42, no. 8, pp. 30-37.

Li, H.; Wu, D.; Tang, W.; Mamoulis, N. (2015): Overlapping community regularization for rating prediction in social recommender systems. *ACM Conference on Recommender Systems*, pp. 27-34.

Li, X.; Zhang, R.; Li, J. (2017): User interest propagation and its application in recommender system. *IEEE 29th International Conference on Tools with Artificial Intelligence*, pp. 218-222.

Ma, H.; Yang, H.; Lyu, M. R.; King, I. (2008): SoRec: Social recommendation using probabilistic matrix factorization. *ACM Conference on Information and Knowledge Management*, pp. 931-940.

Ma, H.; King, I.; Lyu, M. R. (2009): Learning to recommend with social trust ensemble. *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 203-210.

Ma, H.; Zhou, D.; Liu, C.; Lyu, M. R.; King I. (2011): Recommender systems with social regularization. *ACM International Conference on Web Search and Data Minin*, pp. 287-296.

Massa, P.; Avesani, P. (2004): *Trust-Aware Collaborative Filtering for Recommender Systems.* Springer Berlin Heidelberg.

Qian, X.; Feng, H.; Zhao, G.; Mei, T. (2014): Personalized recommendation combining user interest and social circle. *IEEE Transactions on Knowledge and Data Engineering*,

vol. 26, no. 7, pp. 1763-1777.

Ricci, F.; Rokach, L.; Shapira, B. (2015): *Recommender Systems Handbook.* Springer, USA.

Salakhutdinov, R.; Mnih, A. (2007): Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, pp. 1257-1264.

Su, X.; Khoshgoftaar, T. M. (2009): A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, vol. 2009, no. 4, pp. 1-19.

Wang, X.; Liu, G.; Pan, L.; Li, J. (2016): Uncovering fuzzy communities in networks with structural similarity. *Neurocomputing*, vol. 210, pp. 26-33.

Wang, X.; Liu, G.; Li, J.; Nees, J. P. (2017): Locating structural centers: a density-based clustering method for community detection. *PLoS One*, vol. 12, no. 1, pp. 1-23.

Xu, R.; Wunsch, D. (2005): Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645-678.

Yang, J.; Leskovec, J. (2013): Overlapping community detection at scale: a nonnegative matrix factorization approach. *ACM International Conference on Web Search and Data Mining*, pp. 587-596.

Yang, X.; Guo, Y.; Liu, Y.; Steck, H. (2014): A survey of collaborative filtering based social recommender systems. *Computer Communications*, vol. 41, no. 5, pp. 1-10.

Yang, X.; Steck, H.; Liu, Y. (2012): Circle-based recommendation in online social networks. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1267-1275.