What You Like, What I Have: Online Dating Recommendation via Matching Individual Preferences with Features

Xuanzhi Xheng, Guoshuai Zhao, Li Zhu, and Xueming Qian

Abstract—Dating recommendation becomes a critical task since the rapidly development of the online dating sites and it is beneficial for users to find their ideal relationships from a large number of registered members. Different users usually have different tastes when choosing their dating partners. Therefore, it is necessary to distinguish the user's personal features and preferences in dating recommendation methods. However, present approaches don't capture enough user preferences from social graph and attribute data. They also ignore user attributes, which is the complementary and consistent side information of user social graphs. In this paper, we propose a Matching Individual Preferences with Features (MIPF) model to recommend dating partners jointly using user attributes and social graphs. We aim to model user features and preferences to identify what user has and what user likes. We also distinguish user preferences into explicit preference and implicit preferences. The implicit preferences are mined from social graphs, while the explicit preferences are captured from the social links. Additionally, convolutional neural networks are used to extract the latent non-linear information in user attributes. Experiments on real-world online dating datasets demonstrate our MIPF model is superior to existing methods.

Index Terms—Recommender system, Dating recommendation, Preference embedding, Social network.

1 Introduction

Online dating sites such as ShiJiJiaYuan[1], Match.com[2], EliteSingles[3], have developed rapidly and have a large number of registered members. According to a recent survey [1], nearly 40 million single people (out of 54 million) in USA have signed up with Match.com, eHarmony[4] and other online dating sites to seek potential romance. About 20% of committed relationships began online, which is more than through any means other than meeting through friends. In our case, ShiJiJiaYuan is the most popular online dating site in China. There are more than 17 million registered users with rich personal information.

Users upload their self-introductions and then seek for serious relationships. They can follow other users to track their social dynamic. This interaction between users constitutes the user social networks. There are too many registered users on the dating sites. As a result, it is impractical for users to find their right people from a huge number of online users. Dating recommendation methods are frequently implemented to alleviate information overload. A well-performed dating recommendation method is what online dating sites desperately need. In the related research about online dating, many works focus on data mining, user behavior analysis and mate preference modeling. Hitsch et al. [2] studied the economic mechanisms underlying match formation and the formation of marriages using an online dating dataset. They estimated a model of mate preferences and used the Gale-Shapley algorithm to approximate the observed online matching patterns. Su et al. [3] attempted to differentiate the gender-specific mate preferences and reveal the factors affecting potential mate choice by analyzing the user behavioral data of a large online dating site. Kleinerman et al. [4] introduced and evaluated the use of "reciprocal explanations" for explanation method in in reciprocal environments includintg the online-dating domain. For online dating recommendation task, the most of existing methods are based on collaborative filtering approaches. Akehurst et al. [5] reformed the content-collaborative method for reciprocal dating recommendation. The approach used similarity

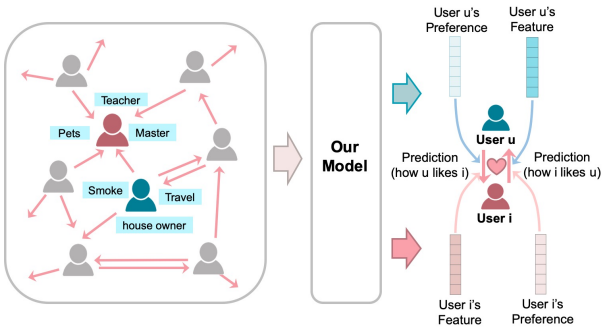


Fig. 1: An illustrative example of MIPF's working principle. User's feature and preference embeddings are computed with the social networks and user attributed information.

- *X. Zheng, and X. Qian are with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, and the SMILES LAB, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: zhengxuanzhi@stu.xjtu.edu.cn, qianxm@mail.xjtu.edu.cn.*
- *G. Zhao and L. Zhu is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: guoshuai.zhao@xjtu.edu.cn, zhuli@mail.xjtu.edu.cn.*

1. https://www.jiayuan.com
2. http://cn.match.com
3. https://www.elitesingles.com
4. http://www.eharmony.com/

between user profiles and the interaction history of the similar users to produce the final recommendations.

In friend recommendation, which is a similar task to dating recommendation, multiple methods are used to predict unknown friendship, including matrix factorization, collaborative filtering, and deep neutral network. SoRec [6] is a probabilistic matrix factorization approach for Social Recommendation. This method employed the shared user latent factors to solve the data sparsity and poor prediction accuracy problems. Rendle et al. [7] transformed the friend recommendation problem into a rank task. They presented a generic optimization criterion and a learning algorithm for personalized ranking. Zhao et al. [8] proposed a collaborative-filtering model calculating the taste and attractiveness of users to predict the stable matches.

The existing dating recommendation methods face two main challenges: lacking of comprehensive user preference modeling and an appropriate way to fuse social graphs and attributes data. Distinguishing user features and preferences may be beneficial to improving the performance of personalized dating recommendation. From our perspectives, compared with the high similarity of user features, the match between preferences and features among users is more important when measuring the matching degree of users in dating recommendation. At the same time, fusing two different modalities, social graphs and attribute information, is beneficial for model to achieve high-accuracy performance.

In this paper, we design the MIPF method for dating recommendation which defines user feature embeddings to model the user individual information and user preference embeddings to depict the characteristics user prefer. We mine the explicit and implicit preference from explicit social links as adjacency matrices and social graphs. The latent topology structures in social graphs are encoded as the implicit preference of users. To enhance the learning of user representations, we use convolutional neural networks to learn the non-linear latent patterns in user attributes and employ graph implicit constraint module to capture the higher order proximity information of the input relation networks. We conduct experiments and compare MIPF with several state-of-art baseline methods on datasets collected from ShiJiJiaYuan dating site. Empirical results demonstrate the effectiveness and rationality of MIPF. The contributions of this paper are as follow:

- We propose a Matching Individual Preferences with Features (MIPF) model for online dating recommendation. Our model fuses user attributes and social graphs to learn user feature and preference representations respectively.
- We divide the user preferences into implicit and explicit preferences. The implicit preferences are mined from social graphs, while the explicit preferences are captured from the social links.
- Additionally, we use two CNNs to extract the latent information in user attributes. The statistical results on datasets prove the vital role of user attributes in recommendation. Experiments demonstrate our model significantly outperforms existing methods.

2 Related Work

In this section, we review several studies about dating recommendation, and then we briefly introduce the recent research focused on social graph representation learning in recommender systems.

2.1 Dating Recommendation

The researches about online dating are more focused on data mining, user behavior analysis and mate preference modeling. Most of the existing online dating recommendation methods are traditional machine learning approaches, such as matrix factorization. The deep learning architecture which has proved to be effective in the recommender systems has not been used.

Collaborative filtering transforms the relationship between users into a real-valued rating matrix, and utilizes collaborative filtering approaches to predict the probabilities of unknown relationships. Zhao et al. [8] proposed a new collaborative-filtering model which considered taste of users in picking others and their attractiveness in being picked by others in bipartite and reciprocal social networks. Akehurst et al. [5] proposed CCR, a new content-collaborative reciprocal recommender. It calculated similarity between user profiles to find users who are similar to the target user. The interaction history of similar users helped the collaborative filtering method to produce the final recommendations. Li and Li [9] analyzed the characteristics of reciprocal online dating recommendations. They proposed a generalized reciprocal recommendation framework which captured user mutual preferences and bipartite relation networks. Xia et al. [1] presented a recommendation system and inteoduced a similarity measure that captured the unique features of the online dating networks. The results of their algorithm on a real-world dataset showed the collaborative filtering-based algorithms outperformed than content-based algorithms in dating recommendation. Since the dating recommendation datasets may contain the textual information of users and their interaction message, natural language processing methods can also be introduced into the dating recommendation field to extract latent features of textual data. Tu et al. [10] designed a two-side matching framework for online dating recommendation. It contained a Latent Dirichlet Allocation [11] to learn the user preferences from the observed user messaging behaviors and user profile features.

Friend recommendation which is similar to the dating recommendation task has attracted much attention in the past decade. The methods of friend recommendation can be divided into three categories: classification, fitting, and ranking [12]. The classification methods shift friend recommendation problems into binary classification problems and train a classifier to predict the unknown friendships. Typical classification approaches such as support vector machine, decision tree classifier and logistic regression have been applied to recommendation task. Benchettara et al. [13] employed a decision tree classifier to make the predictions of the co-authorship likelihood. Gong et al. [14] predicted new and missing links in Google+ using support vector machine.

Fitting methods [15] [16] regard the friend relationships between users as a rating matrix, and predict the likelihood of unknown friendships though matrix factorization [17] [18] and other approaches. Although many existing CF methods mainly focus on the rated items and the context of users to improve recommendation performance, Yang et al. [19] only used the user-item rating matrix without any additional information to make recommendations accurately, efficiently and serendipitously. They designed a novel method, CSII, to tackle the unrated items and find the candidate item sets. The item sets included the latent interesting items, latent satisfying items and serendipitous items of users. Zhang et al. [20] proposed a Kernel-based Attribute-aware Matrix Factorization model called KAMF which exploited the rich attribute information and social links of users to alleviate the rating sparsity effect and tackle the cold-start problems. Zhao et al. [21] proposed a unified matrix-factorized framework with considering four social network factors to predict user-service rating. He et al. [22] proposed a topic community-based method via Non-negative Matrix Factorization (NMF). NMF constrained the two factorized matrices to be non-negative and had a better performance compared with MF in non-negative data.

Both the friend recommendation datasets and the dating recommendation datasets are imbalance between the amount of observed relationships and the unobserved ones. This causes the classification or fitting methods to be biased towards negative prediction. Hence, the ranking methods have been introduced to solve the imbalance issue. The ranking methods calculate the probability of friendships for both observed and unknown relationships. The model attempts to predict the probability of observed relationships larger than that of the unknown or negative relationships. Bayesian Personalized Ranking is the maximum posterior estimator derived from a Bayesian analysis of the problem. Krohn-Grimberghe et al. [23] and Rendle et al. [7] integrated Bayesian Personalized Ranking (BPR) with matrix factorization and showed that BPR provided superior results in combating the imbalance issue.

Recently, deep neural network (DNN) has achieved success in many fields such as image classification, natural language processing and data mining. Compared with traditional machine learning methods, DNN like convolutional neural network with properly designed structure and careful parameter optimization shows superior advantages. Based on the superiority of DNN architecture, several works [24] [25] [26] introduced DNN into the recommendation field, and it has proved to be effective. Ding et al. [25] proposed a BayDNN model by combining Bayesian Personalized Ranking and Deep Neural Networks. It extracted latent structural patterns from the input network data and used the Bayesian ranking to make friend recommendations. Rafailidis et al. [26] proposed a deep pairwise learning model based on a Bayesian personalized ranking strategy for friend recommendations. The deep pairwise network learned the non-linear deep representations of the location-based social networks.

2.2 Social Network Representation Learning

There are many studies about learning low-dimensional representations of nodes in networks. Unsupervised representation learning approaches typically exploit the spectral properties of various matrix representations of graphs, especially the Laplacian and the adjacency matrices. Several dimensionality reduction techniques have been proposed [27] [28] [29]. However, these methods suffer from both computational and statistical performance drawbacks. The expense of matrix eigendecomposition limits these methods to scale to large networks. Moreover, the optimization objectives of these methods are not robust to be applied to other networks with different structures.

Inspired by the recent advancements in representational learning for natural language processing, simulation random walk methods [30] [31] [32] explore a new way for discrete node feature learning. These methods encode the random walk transition matrices and minimize the distances of node embeddings if they are neighbors in the input graphs. Perozzi et al. [33] proposed a novel approach for learning latent social representations. It used random walks to generate node sequences from graphs, then treated them as sentences though Skip-Gram. Node2vec [34] learned a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes. It combined breadth-first sampling and depth-first sampling. Instead of performing simulated walks on the networks, LINE [35] designed a clear objective function to optimize both first-order and second-order graph proximities.

Deep neural networks (DNN) are well known for their advantages in non-linear function learning. Wang et al. [36] introduced DNN to capture the highly non-linear topology structures of networks. Abu-El-Haija [30] proposed a novel objective function, named Graph Likelihood. This approach explicitly modeled the structural information captured from sampled random walks as a function of node representations. However, these random-walk embedding works only investigate the topological structure. Attributed network embedding methods [37] [38] [39] that takes both attribution and relational information into account have been proposed. Zhang et al. [40] proposed a neighbor enhancement autoencoder and attribute-aware skip-gram model named ANRL to capture network structures and node attribute information. Huang et al. [31] used random walk on attributed networks, and implemented a graph neural network architecture to conduct attributed network embeddings.

3 METHODOLOGY

Problem Definition. Our method aims to recommend dating partners for users in online dating sites. The user relationship network can be naturally represented as a homogeneous attributed network $G = (V, E, X)$, where V is a set of nodes representing users, E is a set of directed edges denoting the relationships between users and X is a matrix that encodes all user attribute information. Our aim is to represent each user $i \in V$ as two low-dimensional vectors $f_i \in \mathbb{R}^d$, $p_i \in \mathbb{R}^d$ as user feature and preference embeddings calculated from social graphs and links. d is the dimension of the learned embedding. In this paper, we use capital variables (e.g., A) to denote matrices and lowercase variables (e.g., a) to denote row vectors. For example, we use a_i to mean the ith row of the matrix A, and a_{ij} to

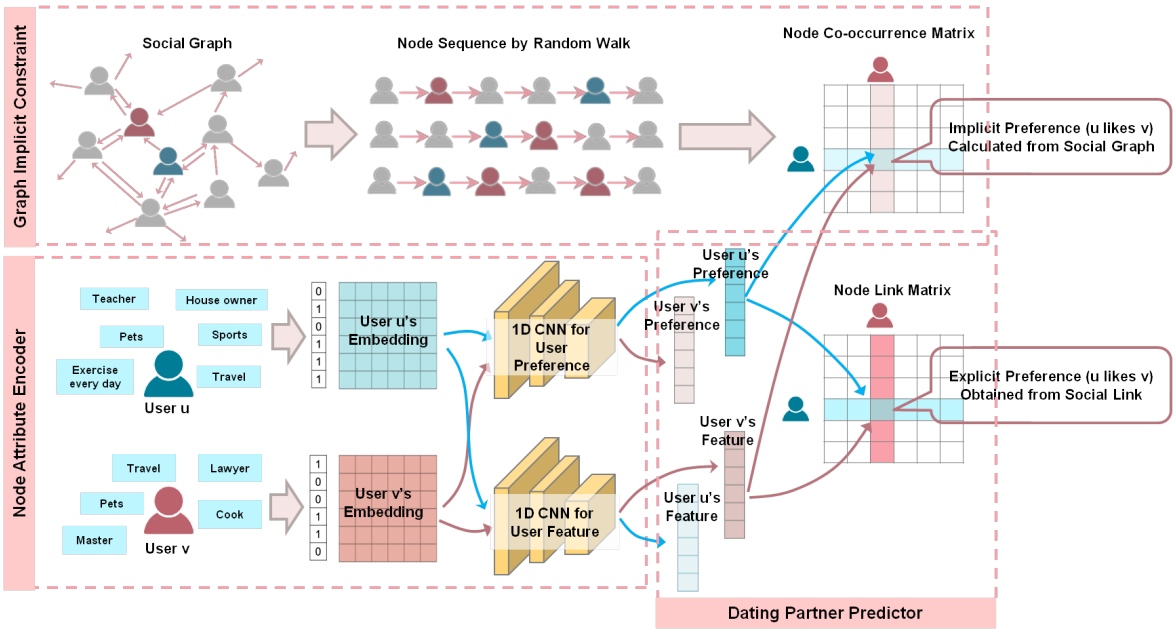


Fig. 2: The overview of our proposed model MIPF

refer to the element of ith row and jth column of the matrix A.

In this section, we will introduce how to learn user features and preferences which eventually identify what user has and what user like. We will show our model in four aspects: 1) Dating partner predictor, which is used to predict the score of dating candidates. 2) Node attribute encoder, which aims to learn user explicit preferences and conduct node embeddings. 3) Graph implicit constraint, which aims to capture user implicit preferences from topological structure and represent them in node embeddings. 4) Loss function, which fuses the loss function of graph implicit constraint and the prediction loss function to train all modules simultaneously. The architecture of our approach is shown in Figure 2.

3.1 Dating Partner Predictor

The dating partner predictor module aims to predict the score $s_{ui} \in [0, 1]$ of user u and user i according to their feature and preference embeddings. The score is the probability of that u will take interested in i.

To recommend candidate user i to user u as a dating partner, we verify whether user u likes the features of user i or not. The match score is calculated by the inner product of the preference embedding p_u of user u and the feature embedding f_i of i:

$$s_{ui} = p_u \cdot (f_i)^T \tag{1}$$

Furthermore, the match score matrix among all users denotes S. We can clearly see that s_{ui} is different from s_{iu} both in the value and the latent meaning.

3.2 Node Attribute Encoder

The role of node attribute encoder module is to learn the user's feature and preference embedding from his attribute information. The attribute information of user u will be onehot processed as the input $x_u \in \{0, 1\}^n$, where n is the length of processed input. The output of node attribute encoder is the feature embedding $f_u \in \mathbb{R}^d$ and preference embedding $p_u \in \mathbb{R}^d$ of user u. Two convolutional neural networks are used to capture the latent structural patterns from the input data. CNN is an effective neural architecture for feature mapping. It is widely used in visual recognition, target detection, pattern recognition and so on. We introduce 1D CNN in our model which is often used in sequence processing and natural language processing task, and it has been considered quite effective. Two CNNs are used to enhance the learning of user feature and preference embeddings respectively. They have the same structure but won't share parameters. We will show details of the feature encoder CNN.

Denote the input attribute data of user u as x_u, the result of the convolutional layer c is calculated as:

$$c = tanh(W * x_u + bias) \tag{2}$$

where the W and bias denote the parameters of CNN

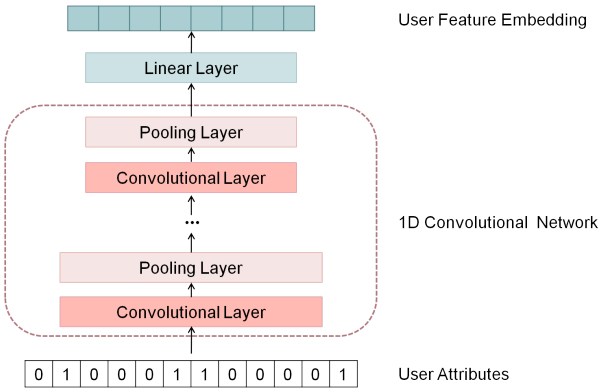


Fig. 3: The detail of node attribute encoder

filters. $tanh$ is used as the non-linear and non-negative function for activation. The results of the convolutional layer

will be sent to the pooling layer to aggregate features and reduce the dimension. We use max function to implement pooling operation, and the outputs of the pooling layer can be written as:

$$h = pool(c) \tag{3}$$

In order to enhance the ability of node attribute encode to capture the high-level structural patterns in user attribute data, we repeat the combination of a convolution layer and a pooling layer three times. The input of each convolutional layer is the output of the last pooling layer. We denote l as the l-th layer of the node attribute encoder. For the l-th layer, we have:

$$h^l = pool(tanh(W^l * h^{l-1} + bias^l)) \tag{4}$$

After three times loop of the convolution layer and the pooling layer, the dimension of output vectors is shorter than that of user feature embedding we expected. We add a linear layer and anactivation function to transform the vector dimension. The final feature embeddings can be written as:

$$f_u = ReLU(W^{l+1} * h^l + bias^{l+1}) \tag{5}$$

3.3 Graph Implicit Constraint

The graph implicit constraint module mins the latent topological information contained in social graphs. It fuses the latent topological information into user preference embeddings as implicit preferences. We adopt simulation random-walk approaches in the graph implicit constraint module to learn the representations of nodes. Simulation random-walk approaches are inspired by recent advancements in learning embedding vectors for words. They sample many random walks from graphs. If two nodes are frequently close in the random walks, they would have large similarity in the embedding space. Different from the previous methods, Abu-El-Haija et al. [30] associated edges with a function of node embeddings and proposed a novel objective function, the graph likelihood, which contrasted statistics from random walks with non-existent edges. They used the node co-occurrence matrix $D \in R^{|V|\times|V|}$ to represent the correlation between two nodes. D recorded the times of co-visited between pairs of nodes in all stochastically random walks on the graph. The graph likelihood objective is improved from the Maximum Likelihood Estimate of Logistic Regression:

$$\max \prod_{(u,i)\in E} s_{ui} \prod_{(u,i)\notin E} (1 - s_{ui}) \tag{6}$$

We define that the indicator function $\mathbb{1}[.]$ evaluates to 1 if its boolean argument is true. The adjacency matrix $A \in \{0,1\}^{|V|\times|V|}$ can be constructed according to $a_{ui} = \mathbb{1}[(u,i) \in E]$. The Equation 6 can be equivalently written as:

$$\max \prod_{u,i\in V} s_{ui}^{\mathbb{1}[(u,i)\in E]} (1 - s_{ui})^{\mathbb{1}[(u,i)\notin E]} \tag{7}$$

To improve generalization of link-prediction, the range of neighbor-sets of nodes is extended beyond the set of node direct connections via random walks according to recent studies. The binary edge presence $\mathbb{1}[(u,i) \notin E]$ can be replaced by simulated random walk statistic matrix D. The graph likelihood can be formulated as:

$$\max \prod_{u,i\in V} \sigma(s_{ui})^{d_{ui}} (1 - \sigma(s_{ui}))^{a_{ui}} \tag{8}$$

where the σ denotes sigmoid activation function. The graph likelihood objective function pushes the model score s_{ui} close to 1 if d_{ui} is large and pushes it towards 0 if there is no edge between node u and i.

However, there are many hyper-parameters in random walks algorithm, which have to be manually tuned for every graph. Graph Attention Models [41] replaced the fixed hyper-parameters with trainable ones which were learned in training process automatically. In particular, Graph Attention Models added attention mechanism to get analytical expression for $E[D]$, the expectation of co-occurrence matrix D. After replacing D with $E[D]$ and taking the negative log of the graph likelihood for avoiding quadratic computation, the Negative Log Graph Likelihood (NLGL) can be written as:

$$\min \sum_{u,i\in V} -E[d_{ui}] log(\sigma(s_{ui})) - a_{ui}(1 - \sigma(s_{ui})) \tag{9}$$

In our proposal, the matrix $E[D]$ between users is calculated to capture the structural information. NLGL loss is introduced as a part of objective function which is an implicit constraint for user preference embedding learning.

3.4 Loss Function

In our method, the social links between users can be regarded as the user explicit preferences, while the potential structural information hidden in the social graph is considered as the implicit preferences. In the Graph Implicit Constraint module, our model mines user implicit preferences with the constraint of the NLGL loss function. We propose prediction loss to learn the user explicit preferences.

For each user u, the other users can be divided into three disjoint sets, a set with positive dating partners $P_u = \{i | i \in V, (u,i) \in E\}$, a set with negative dating partners N_u and a set of unobserved users K_u. In addition, $P_u \cup K_u = \{i | i \in V, (u,i) \notin E\}$. The positive user set and a part of negative set are constructed according to the user interaction records in datasets, and the other part of negative user set is generated through negative sampling. The remaining users have unknown relationships with user u, thus we classify them as unobserved users K_u. The matrix $M \in \{-1,0,1\}^{||V||*||V||}$ can be defined according to $m_{ui} = 1$ when $i \in P_u$, $m_{ui} = 0$ when $i \in K_u$, and $m_{ui} = -1$ when $i \in N_u$. We will explain the details of how we divide the user sets and how we implement the negative sampling in EXPERIMENTS section.

The prediction loss is to maximize the score of positive dating partners and minimize that of negative dating partners. It can be formulated as :

$$\mathcal{L}_p = -\|\sigma(S) \circ M\|_1 \tag{10}$$

where $\circ$ is the Hadamard product, the function $\|\cdot\|_1$ of a matrix is the sum of its entities. To combined with the

graph implicit constraint modules, we rewrite our NLGL loss function as :

$$NLGL = -\|E[D] \circ log(\sigma(S)) + \|A - 1\| \circ log(\sigma(1 - S))\|_1 \quad (11)$$

Our final objective is to minimize the sum of the prediction loss and NLGL loss:

$$\mathcal{L} = -\alpha\|\sigma(S) \circ M\|_1 - \|E[D] \circ log(\sigma(S)) + \|A - 1\| \circ log(\sigma(1 - S))\|_1 \quad (12)$$

where α is the weighting coefficient to balance the prediction loss and NLGL loss. We adopt the stochastic gradient algorithm for optimizing Equation 12. We iteratively optimize all the parameters until the model convergency.

4 EXPERIMENTS

4.1 Datasets

The two datasets we used are from the ShiJiJiaYuan website (https://www.jiayuan.com/), one of the most famous online dating sites in China. Up to now, it has more than 300 million users. Compared with other online dating websites, it has a large number of users with comprehensive and effective user profiles. Moreover, the interaction records between users are showed on the website, which is the basic of constructing user social graphs. While most of other online dating websites do not revealed the social relations of users.

4.1.1 ShiJiJiaYuan Data Mining Competition Dataset (SDMCD)

The dataset is provided by ShiJiJiaYuan Company for the first data mining competition for college students[56]. It contains anonymized profiles and heterosexual dating interaction records of more than 50,000 users in city Hangzhou during a period of 3 months. It contains 344,522 male users and 203,843 female users. Each user owns 34 personal features, which are showed in detail in table 2. The above profiles are filled in by users according to their wishes. If some fields in profiles that users are unwilling to fill in, it will be showed as "inconvenient to disclose". After statistics, the average number of valid fields for each user is 17.22.

The dataset contains over 8,600,000 interaction records. The user interaction records can be divided into three categories: recommendation, click, and message. Recommendation records represent that the system recommended a user to another user. Click records indicate that a user clicked and browsed the homepages of the user recommended by the system. Message records indicate that a user messaged to other users after browsing their homepage. For instance, the system recommends user B and C to user A when A visits the ShiJiJiaYuan site. A is interested in B and clicks to view the homepage of B. After that, A has further interest and sends messages to B. There are three interaction records between A and B belonging to recommendation, click, message categories respectively, and a record between A and C belonging to recommendation category. The proportion of these each category to the total interaction records is shown in Table 1.

5. https://github.com/Zhengbaibai/ShiJiJiaYuan-Competition-Dataset

6. https://cosx.org/2011/03/1st-data-mining-competetion-for-college-students/

TABLE 1: The proportion of three categories of interaction records in SDMCD.

Categories	Recommendation	Click	Message
Proportion	97.29%	2.14%	0.57%

We compute the conversion probability between these three categories. The results are shown in the figure 4. Only 2.2% of user pairs with recommendation records have the subsequent click records. The user pairs with click records have 27% chance to perform message action, which much higher than the probability from recommendation records to click records. Hence, we regard user pairs who only have recommendation records as wrong recommendations. The click records and the message records can be merged as correct recommended user pairs. Considering the training time

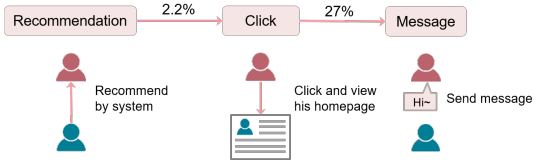


Fig. 4: The probability of having subsequent actions

and the runtime memory requirements of our method and baselines, we use a random selection method to generate a small dataset with 15,669 users. The profiles of users are completely retained. For the interaction records, they are retained if the users in records are all selected in the small dataset. There are 180,816 interaction records was reserved in the final small dataset.

4.1.2 ShiJiJiaYuan Crawled Dataset(SCD)

The dataset is crawled from the ShiJiJiaYuan site. It contains 11,069 anonymous active user profiles and 20,419 their interaction records which are all open to registered users. The dataset has 3,146 male users accounting for 28.04% and 7,923 female users. Each user profiles have 64 fields. Some of fields are required for user such as age, sex, marital status, and so on. Other fields like jobs, fitness frequency, education level and family members can be omitted according to the user's willing. The user relationships in SCD are the records of following action in the ShiJiJiaYuan site. Users can follow other users and track their newest social dynamics. It should be noted that a following action denotes a direct relationship.

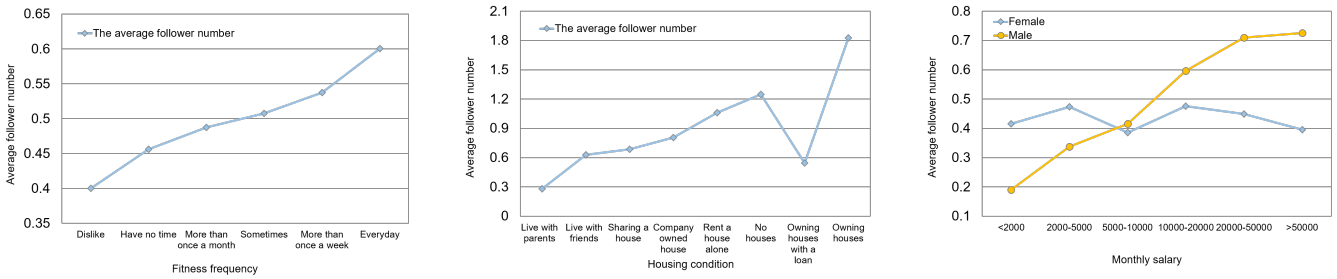
Moreover, we conduct a statistical analysis on SCD to study the effect of user attributes in selecting dating partners. Firstly, we count the average follower number of user groups divided by different fitness frequency. The follower number represents the user popularity in dating market. In figure 5(a), it is shown that the follower number raises with the increasing fitness frequency. It means users tend to pick the dating partners who are keen on fitness. Secondly, we explore the relationship between user popularity and their housing conditions. From figure 5(b), we can get the conclusion that the popularity of users is closely related to their housing conditions. People owning a house are more popular in the dating market. Living with parents or friends will decreases their attractiveness. In addition, we find that there is difference between men and women's preferences in

TABLE 2: Statistics of Datasets.

Dataset	#User	#Social relation	#Profile fields	#Average filled fields	#Male percentage	#Female percentage
SDMCD	548,365	8,600,000	21	17.22	62.8326%	37.1673%
SCD	11,069	20,419	64	38.82	28.0427%	71.9573%

TABLE 3: self-description features in ShiJiJiaYuan Crawled Dataset.

scalar		categorical						
age height	match_age match_height	sex pet salary location	zodiac nation children siblings	smoking drinking housework education	car_status blood_group schedule house_status	occupation constellation company_type marriage_status	match_belief match_education match_location registered_residence	exercise_habit live_with_parents match_marriage_status



(a) Fitness frequency analysis. (b) Housing condition analysis. (c) Monthly salary analysis.

Fig. 5: Average user follower number with different attributes on SCD.

some specific attributes such as monthly income, showed in figure 5(c). Men with high monthly salary are more popular in dating market, while the different income of women makes no difference. These statistical results evidently confirm that user attributes contain abundant information and play a vital role in modeling user representations.

4.2 Comparison Methods

In recent years, most of dating recommendation works use non-public datasets with additional information, which makes it impossible for us to reproduce and compare them with MIPF on SCD and SDMCD. Therefore, we compare MIPF with two types of state-of-the-art methods, covering friend recommendation methods and network embedding algorithms. Friends recommendation methods contain fitting approaches, FM [42], and ranking approaches, BPRMF [43] and BayDNN [25]. Four network embedding methods, LoNGAE [44], PRRE [45], NGCF [46], Graph Attention Model [41], are also trained to generate recommendations by computing the cosine similarities between users.

1) **FM** [42]: A basic factorization machine for prediction problem, which can mimic most factorization models just by feature engineering. Menon and Elkan et al. [47] proposed to solve the link prediction problem by using factorization machines and regarding positive links as 1 and the others as 0.
2) **BPRMF** [7]: BPRMF is a matrix factorization model relying on bayesian loss optimization, which is the maximum posterior estimator derived from a Bayesian analysis of personalized ranking. Both theoretical and empirical results indicate the effectiveness of BPRMF for the field of personalized ranking.
3) **LoNGAE** [44]: An autoencoder architecture for link prediction, which has the capability to learn expressive non-linear latent node representations from both local graph neighborhoods and explicit node features.
4) **PRRE** [45]: PRRE is an attributed network embedding method to exploit the partial correlation between node topology and attributes, which affects the actual proximity among nodes in the embedding space.
5) **BayDNN** [25]: BayDNN combines Bayesian Personalized Ranking and Deep Neural Network for friend recommendation. A convolutional neural network is integrated to extract latent deep structural feature representations. And a Bayesian personalized ranking learning algorithm is used to captures user preferences based on the extracted deep features.
6) **NGCF** [46]: Neural Graph Collaborative Filtering (NGCF) is a new recommendation framework, which integrates the user-item interactions with the embedding learning process in the form of a bipartite graph and exploits the graph structure by propagating embeddings on it. By leveraging the expressive modeling of high-order connectivity in user-item graph, NGCF effectively injects the collaborative signal into the embedding process in an explicit manner.
7) **Graph Attention Model** [41]: Graph Attention Model is a state-of-the-art graph embedding method, which learns the context distribution in graph embedding methods via an attention mechanism, and replaces the hyper-parameters with trainable models to learn the best suited hyper-parameters for graphs automatically. It has been shown to achieve substantially improvement on link prediction tasks.

TABLE 4: Performance comparison on AUC and AP.

Methods	SCD					SDMCD				
	AUC	AP	@P1	@P5	@P10	AUC	AP	@P1	@P5	@P10
PRRE	0.592308	0.420605	0.070596	0.318711	0.619945	0.541793	0.447266	0.042764	0.261150	0.574055
BPRMF	0.588795	0.641575	0.189856	0.410212	0.606237	0.678500	0.715266	0.260270	0.544129	0.699794
LoNGAE	0.646378	0.617912	0.187966	0.344089	0.693882	0.690565	0.648130	0.204657	0.389631	0.652046
FM	0.654159	0.688491	0.275873	0.520904	0.706648	0.757021	0.747234	0.329358	0.545009	0.667346
BayDNN	0.717075	0.702463	0.288608	0.644619	0.854982	0.713433	0.728658	0.264152	0.630950	0.798525
NGCF	0.784034	0.781208	0.269886	0.647727	0.809659	0.833769	0.827644	0.230078	0.755176	0.860938
Graph Attention Model	0.804417	0.824549	0.530500	0.759424	0.870802	0.819113	0.837645	0.461287	0.785311	0.903426
MIPF(Ours)	**0.831626**	**0.859338**	**0.558602**	**0.767649**	**0.875600**	**0.852843**	**0.859331**	**0.467030**	**0.809847**	**0.905152**
%Improv.	3.27%	4.21%	5.29%	1.08%	0.55%	2.28%	2.58%	1.24%	3.12%	0.19%

4.3 Implementation

The user profiles in the two datasets contains various fields, which is comprehensive description of users. However, some fields are omitted by most of users. Some fields are free text, which is difficult to process as the scalar and categorical data. Considering sparse attribute representations will increase memory usage and decrease the model performance, we conduct a preliminary analysis of these attributes and delete some fields manually according to their importance and sparsity. There are 21 and 16 fields reserved in SDMCD and SCD. We perform One-Hot processing to convert the Chinese categorical options and numeric data in user profiles into vectors containing only 0 and 1.

As user interaction records, we use negative sampling techniques. In SDMCS, each user who is clicked or sent messages by other users can be regarded as a positive sample. We randomly sample K users which are not interacted with this user or only recommended to this user. In SCD, we regard a user which is followed by a user as a positive sample. And we also randomly select K users as negative samples from all users except positive users. Then we rank the candidate user list which consists of one positive user and K negative users. We minimize the loss function Equation 12 to make the model rank positive samples higher than negative samples. In our experiments, the length of candidate user lists is 20. K in training set is 1 and in testing set is 19. For MIPF and baselines, we randomly select 80% positive samples for the training set, and 20% positive samples for the testing set. To ensure all users appear in training set, in the process of dividing the training and testing set, we first check whether users in interaction records have appeared in the training set. If not, they will be directly classified into the training set, otherwise they will be divided randomly.

For matrix factorization methods, FM and BPRMF, we use interaction records by regarding the positive samples as 1 and negative samples as 0. For graph embedding methods without considering node attributes, such as Graph Attention Model, NGCF and BayDNN, they only use social graphs as input and use CNN or randomly walk algorithms to learn user representations. The prediction of dating partners is generated though the inner product of the user representations. For attributed graph embedding methods, such as LoNGAE and PRRE, they learn user embeddings incorporating node attributes and topology structure of social graphs. The hyper-parameters of comparative methods are used as the default settings in shared source code, include FM, BPRMF, PRRE, LoNGAE, NGCF and Graph Attention Model. The source code of BayDNN does not shared, so we implement it by ourselves and set the initial hyper-parameters according to the paper of BayDNN. The dimension of user embedding is set to 64 and batch size is 32 in BayDNN on both datasets.

In our MIPF method, we use random seed as 66478 to initialize the weight parameters in convolutional neural networks. The weighting coefficient α is 0.0000001 on SCD, and 0.000001 on SDMCD. Dimension of user feature and preference embeddings varies from 8 to 128 depending on the complexity of datasets, specifically, we choose 8 as d on SCD and 128 on SDMCD. Batch size for both two datasets is 32. The experiments are conducted on a machine with one GPU (NVIDIA RTX-2080Ti) and one CPU (Intel Xeon CPU E5-2620).

4.4 Results and Discussions

The experiment results on SCD and SDMCD datasets are illustrated in Table 4. According to Table 4, we have several observations.

First, our approach can consistently outperform all compared baseline methods. On SCD and SDMCD, MIPF outperforms the best baseline Graph Attention Model respectively by 3.27% and 2.28% in terms of ACU. Although Graph Attention Model introduces attention machine in simulation random walk to capture the higher order proximity information of social graphs with trainable hyperparameters, they have a disadvantage that they do not take the side information into account. Our approach designs a node attribute encoder module and the prediction loss function to extract the latent information in user attributes. Thus, our approach achieves better performance. BayDNN method combines deep neural network for feature representation learning and adopt Bayesian personalized ranking to optimize the method. Compared with BayDNN, our approach not only takes advantage of the user attributes but also learns a better social graph structure representation. The superior performance of our approach demonstrates graph structure and node attributes are consistent and complementary information for dating recommendation, and it is important to design an effective graph representation algorithm.

Second, choosing the right graph representation learning algorithm is important to dating recommendation performance. PRRE is an attributed network embedding method to exploit the partial correlation between node topology and attributes. The experiments show that it not a suitable method for dating recommendation cause the attribute data and topology of social graphs in our datasets are

TABLE 5: Ablation study results on AUC, AP and P@n.

Methods	SCD					SDMCD				
	AUC	AP	@P1	@P5	@P10	AUC	AP	@P1	@P5	@P10
BPRMF	0.588795	0.641575	0.189856	0.410212	0.606237	0.678500	0.715266	0.260270	0.544129	0.699794
BPRMF-2	0.596959	0.646679	0.189513	0.411583	0.606923	0.724453	0.769324	0.354204	0.611178	0.735424
MIPF-A-2	0.637188	0.613786	0.315970	0.471899	0.665867	0.668893	0.672681	0.236288	0.569807	0.730399
MIPF-G-2	0.804417	0.824549	0.530500	0.759424	0.870802	0.819113	0.837645	0.461287	0.785311	0.903426
MIPF-AG	0.801323	0.817145	0.526388	0.744688	0.866690	0.819290	0.789203	0.308199	0.778753	0.900617
MIPF-AG-2	**0.831626**	**0.859338**	**0.558602**	**0.767649**	**0.875600**	**0.852843**	**0.859331**	**0.467030**	**0.809847**	**0.905152**

consistent and complementary. LoNGAE is an autoencoder architecture, which learns latent node representations from both graph neighborhoods and node features. However, the performance of LoNGAE is lower than other baselines with only using graph structure information. This might be due to the model is not suitable for our task scenario. BayDNN uses convolutional neural network to extract latent deep structural feature representations, and Graph Attention Model takes advantage of simulation random walk to capture the higher order proximity information of network structure. These two methods both achieve good performance. It has proved that convolutional neural network and random walking are both effective methods to capture topology structure in graphs.

Third, FM and BPRMF are matrix factorization based algorithms. Compared to these traditional methods with shallow architecture or limited levels of feature extraction, deep neural networks like CNN with properly designed structure and careful parameter optimization achieve superior advantages.

4.4.1 Ablation Study.

To demonstrate the effectiveness of distinguish user's feature and preference embeddings, we compare MF and MIPF with single embedding for a user and with feature and preference embedding respectively. We also want to verify that the combination of implicit preference captured from social graph and explicit preference extracted from user adjacency matrix will significantly improve the performance of our MIPF model.

In summary, the experiment configurations are detailed below:

1) **BPRMF**:BPRMF model with single user embedding.
2) **BPRMF-2**:BPRMF model with user feature and preference embeddings respectively.
3) **MIPF-A-2**: MIPF model only uses user attributes as input data.
4) **MIPF-G-2**:MIPF model only uses social graph as input data.
5) **MIPF-AG**: MIPF model with single user embedding and used social graph and user attributes as input data.
6) **MIPF-AG-2**:MIPF model with user feature and preference embedding respectively. It uses social graph and user attributes as input data.

In order to make a fair comparison, the dimensions of user representation vectors in the above experiment are 8 and 128 on SCD and SDMCD. The results of ablation study are showed in Table 5. The performance comparison of BPRMF and BPRMF-2, as well as MIPF-AG, MIPF-AG-2 prove the effectiveness of distinguishing feature and preference embedding in the dating recommendation. The comparison of MIPF-AG-2 with MIPF-A-2 and MIPF-G-2 shows that the AUC value is improved from 63.7% and 80.4% to 83.1% on SCD. The prominent improvement demonstrates user attributes and graph structure information are complementary. The fusion of these two modalities can help our model achieve a relatively significant improvement. When comparing MIPF-A-2 and MIPF-G-2, the performance gap of these two variants is very significant, which indicates social graph are more effective in dating recommendation task. It may due to the fact that user pairs recommended by MIPF-A-2 are difficult to meet on dating sites because of the large number of members. While user pairs recommended by MIPF-G-2 are easier to recognize because of the near distance in social graphs.

4.4.2 Discussion on the hyperparameter α.

We explore the influence of hyperparameter α, which is the weighting coefficient of NLGL loss and prediction loss in Equation 12. Experimental results about hyperparameter α are shown in Figure 6. We can see that α has a high impact on the performance of MIPF. In order to find the optimal value of α, we change the order of magnitude of α and compare the experimental results.

It can be seen that with the increase of α (also the decrease of $log_0.1\alpha$), the values of AUC and AP both lift first, and then gradually reduce after reaching the peak. When α value is small, the proportion of prediction loss in total loss decreases. The model ignores prediction loss which directly related to the recommendation task, and the performance declines. The optimal performance is achieved when the α value is 0.000 000 1 on SCD and is 0.000 001 on SDMCD. At this time, the two loss functions are balanced through α. The model not only makes the score of positive samples greater than negative samples according to the constraint of prediction loss, but also capture the high-order topology information in the social graph through the constraint of NLGL loss. In the continuous increase of α, the prediction loss will be much larger than the NLGL loss. The MIPF model will only take into account the adjacency matrix of the social graph which is the explicit preference of users. The implicit preference of users hidden in high-order information of the social graph will be discarded due to the negligence of the NLGL loss. That is the reason of the lower performance when α is too large.

4.4.3 Discussion on the hyperparameter d .

We explore how the dimension of feature and preference embeddings affects the performance of MIPF on two

TABLE 6: Discussion on the parameter α on SCD and SDMCD.

$log_{0.1}\alpha$	SCD					SDMCD				
	AUC	AP	@P1	@P5	@P10	AUC	AP	@P1	@P5	@P10
2	0.810889	0.821376	0.625086	0.789925	0.889650	0.835156	0.834102	0.406441	0.795197	0.889686
3	0.824909	0.853066	0.529130	0.754969	0.867718	0.849474	0.855622	0.457405	0.807237	0.900062
4	0.826260	0.852570	0.543523	0.761138	0.871145	0.851398	0.856395	0.460472	0.810663	0.905543
5	0.831626	0.859338	0.558602	0.767649	0.875600	0.852033	0.856355	0.461255	0.810532	0.907436
6	0.827130	0.858249	0.554489	0.765593	0.874229	0.852153	0.857100	0.462168	0.810336	0.905772
7	0.827465	0.855631	0.539068	0.767992	0.875942	0.851461	0.855225	0.455708	0.808150	0.907207
8	0.825817	0.855669	0.540781	0.767649	0.871145	0.850583	0.854925	0.467846	0.803028	0.902346
9	0.823996	0. 851258	0.541467	0.759767	0.878341	0.848660	0.851148	0.446736	0.804463	0.906946

TABLE 7: Discussion on the parameter d on SCD.

d	AUC	AP	@P1	@P5	@P10
4	0.830507	0.857030	0.629883	0.776559	0.875257
8	0.831626	0.859338	0.558602	0.767649	0.875600
10	0.823631	0.856584	0.541809	0.761138	0.864633
16	0.807139	0.843367	0.616861	0.793352	0.892735
32	0.803613	0.816053	0.616861	0.793352	0.892735
64	0.804800	0.819789	0.613777	0.789582	0 887252

TABLE 8: Discussion on the parameter d on SDMCD.

d	AUC	AP	@P1	@P5	@P10
32	0.835077	0.841435	0.466638	0791151	0.887533
64	0.852153	0.857100	0.462168	0.810336	0.905772
128	0.852843	0.859331	0.467030	0809847	0.905152
256	0.850059	0.857010	0.467552	0.806878	0.902444
512	0.850707	0.857029	0.467210	0.806388	0.902085
1024	0.846809	0.855965	0.467617	0.795556	0.899409

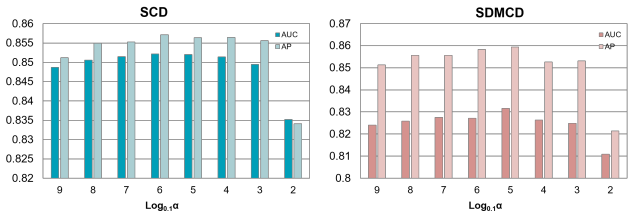


Fig. 6: Performance with different α.

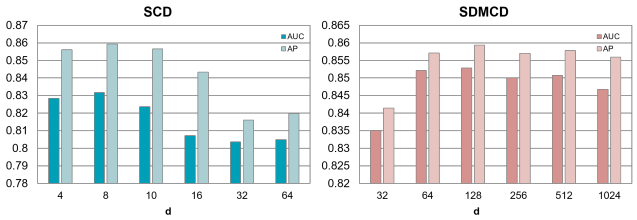


Fig. 7: Performance with different d.

datasets. The experiments results about d are shown in Figure 7. On SDMCD, we can find the performance of AUC and AP shows an upward trend when d raises from 32 to 128. This is probably because the smaller embedding dimension will limit the ability of model to express the information extracted from user attributes and social graphs. Thereby the performance of model recommendation decreases when d is smaller than 128. When d continues growing, the performance of our model declines. It may due to that the model will overfit and lose its generalization ability with excessive dimension of embeddings. Therefore, according to the experimental results, the optimal values of d are 8 and 128 on SCD and SDMCD. The different optimal values of d probably due to the number of interaction records in SDMCD is much larger than that in SCD. User feature and preference embeddings in SDMCD require a larger dimension to express more information extracted from input data.

4.4.4 *Discussion on the hyperparameter K .*

To investigate how the parameter K affects the performance, we vary number of negative samples of each positive sample in the training process. In particular, we try the K value in the range of {1, 2, 4, 8, 16, 32, 64}. Table 9 summarizes the experimental results on both datasets. We can get the conclusion that the performance of MIPF is basically not affected by the change of K value on SDMCD. While on SCD, increasing the value of K will slightly improve the performance. The number of interaction records in SCD is significant less than that in SDMCD. Therefore, increasing the amount of negative samples in the training set may expand the dataset and improve performance on SCD.

4.4.5 *Discussion the performance on user groups with different number of interation records.*

We want to explore the performance of MIPF on data with different sparseness. We compare it with the two baseline methods, Graph Attention Model and NGCF. Users in the testing set are divided into four groups according to the number of user interaction records in the training set. The performance of methods on different user groups shows their applicability on datasets with different sparseness levels. For each user in the testing set, we record the ranking of his positive user in his candidate user list. And we calculate the average ranking of positive users for each user group. The lower average ranking denotes the method has a higher probability to recommend the correct user. Experimental results are shown in Figure 8, our MIPF method can consistently outperform compared baseline methods on user groups with different degrees of sparseness. In addition, MIPF has better performance on user groups with 6-20 interaction records per user. This is probably because when users have too few interaction records, it is difficult to learn accurate user representations. Furthermore, the performance of MIPF deteriorates on user groups with more than 20 interaction records per user. It might because over much social connections will bring noise in the process of graph embedding.

TABLE 9: Discussion on the parameter k on SCD and SDMCD.

k	SCD					SDMCD				
	AUC	AP	@P1	@P5	@P10	AUC	AP	@P1	@P5	@P10
1	0.831626	0.859338	0.558602	0.767649	0.875600	0.852843	0.859331	0.467030	0.809847	0.905152
2	0.829708	0.858485	0.574023	0.771761	0.876628	0.852746	0.859168	0.465464	0.809423	0.905511
4	0.829589	0.856152	0.551405	0.757368	0.868403	0.852499	0.861670	0.475872	0.809619	0.902020
8	0.828249	0.852462	0.558259	0.775531	0.880740	0.852511	0.857941	0.462886	0.810467	0.905739
16	0.830360	0.857315	0.555860	0.766278	0.875257	0.851443	0.854816	0.455382	0.809097	0.907338
32	0.836027	0.862984	0.555517	0.768677	0.871830	0.852704	0.858995	0.465138	0.808738	0.905576
64	0.835078	0.866280	0.556546	0.771761	0.881768	0.852681	0.858735	0.464615	0.808738	0.905543

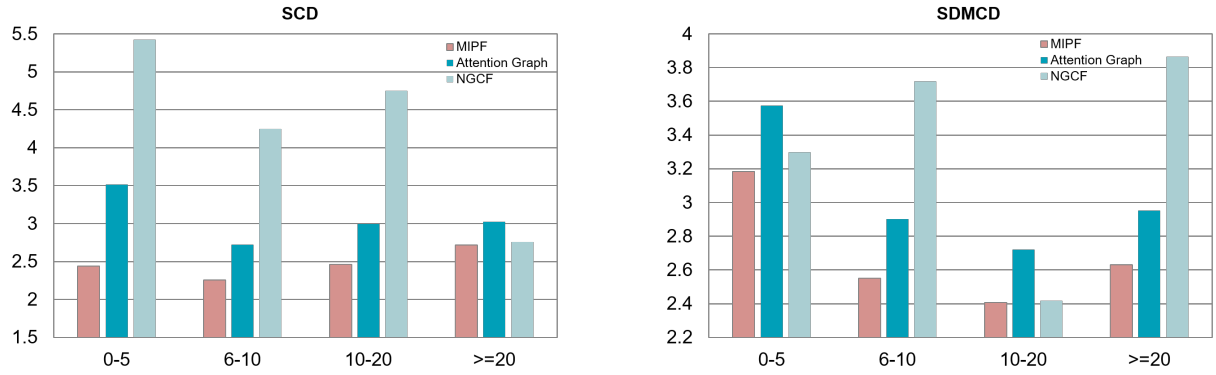


Fig. 8: The Performance of MIPF and two comparison methods on user groups with different number of interaction records on SCD and SDMCD. The y-axis is the ranks of positive users in candidate user lists. The lower average ranking denotes the better recommendation for users.

5 Conclusion

This paper presents an online dating recommendation model MIPF by matching individual preferences with features. It fused user attributes and social graph to conduct user feature embeddings and preference embeddings. The user preference embedding contains explicit preference extracted from social links which is the adjacency matrix of user's relationship network, and implicit preference, the latent topology information in social graph. In order to capture the implicit preference, we use a graph embedding method with attention mechanism guiding the simulation random walk. Experiments demonstrate that our MIPF outperforms comparing methods, and the quantitative results of ablation study verify the effectiveness of our contributions. There are potential future directions of this work. First, we only use scalar and digital attributes, and ignore the free text information such as self-introduction of users. With the help of the latest Natural Language Processing technology, extracting the information in free text can make the model achieve better performance. Secondly, the current graph embedding methods have relatively high requirements for running memories when extracting high-level structural information. Thus, MIPF also has a limit on the scale of datasets, which reduces its scalability.

References

[1] P. Xia, S. Zhai, B. Liu, Y. Sun, and C. X. Chen, "Design of reciprocal recommendation systems for online dating," *Social Netw. Analys. Mining*, vol. 6, no. 1, pp. 32:1–32:16, 2016.

[2] G. J. Hitsch, A. Hortaçsu, and D. Ariely, "Matching and sorting in online dating," *American Economic Review*, vol. 100, no. 1, pp. 130–63, 2010.

[3] X. Su and H. Hu, "Gender-specific preference in online dating," *EPJ Data Sci.*, vol. 8, no. 1, pp. 12:1–12:22, 2019.

[4] A. Kleinerman, A. Rosenfeld, and S. Kraus, "Providing explanations for recommendations in reciprocal environments," in *RecSys*. ACM, 2018, pp. 22–30.

[5] J. Akehurst, I. Koprinska, K. Yacef, L. A. S. Pizzato, J. Kay, and T. Rej, "CCR - A content-collaborative reciprocal recommender for online dating," in *IJCAI*. IJCAI/AAAI, 2011, pp. 2199–2204.

[6] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *CIKM*. ACM, 2008, pp. 931–940.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[8] K. Zhao, X. Wang, M. Yu, and B. Gao, "User recommendations in reciprocal and bipartite social networks-an online dating case study," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 27–35, 2014.

[9] L. Li and T. Li, "MEET: a generalized framework for reciprocal recommender systems," in *CIKM*. ACM, 2012, pp. 35–44.

[10] K. Tu, B. F. Ribeiro, D. D. Jensen, and et al., "Online dating recommendations: matching markets and learning preferences," in *Proc. WWW*, 2014, pp. 787–792.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[12] Z. L. Li, X. Fang, and O. R. L. Sheng, "A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions," *ACM Trans. Management Inf. Syst.*, vol. 9, no. 1, pp. 1:1–1:26, 2018.

[13] N. Benchettara, R. Kanawati, and C. Rouveirol, "A supervised machine learning link prediction approach for academic collaboration recommendation," in *RecSys*. ACM, 2010, pp. 253–256.

[14] N. Z. Gong, A. Talwalkar, L. W. Mackey, L. Huang, E. C. R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint link prediction and attribute inference using a social-attribute network," *ACM TIST*, vol. 5, no. 2, pp. 27:1–27:20, 2014.

[15] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Based Syst.*, vol. 46, pp. 109–132, 2013.

[16] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, 2014.

[17] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[18] C. Chen, X. Zheng, Y. Wang, F. Hong, and Z. Lin, "Context-aware collaborative topic regression with social matrix factorization for recommender systems," in *AAAI*. AAAI Press, 2014, pp. 9–15.

[19] Y. Yang, Y. Xu, E. Wang, J. Han, and Z. Yu, "Improving existing collaborative filtering recommendations via serendipity-based algorithm," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1888–1900, 2018.

[20] J. Zhang, C. Chow, and J. Xu, "Enabling kernel-based attribute-aware matrix factorization for rating prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 798–812, 2017.

[21] Z. Zhao, H. Lu, D. Cai, X. He, and Y. Zhuang, "User preference learning for online social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2522–2534, 2016.

[22] C. He, H. Li, X. Fei, A. Yang, Y. Tang, and J. Zhu, "A topic community-based method for friend recommendation in large-scale online social networks," *Concurr. Comput. Pract. Exp.*, vol. 29, no. 6, 2017.

[23] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *WSDM*. ACM, 2012, pp. 173–182.

[24] C. H. Liu, J. Xu, J. Tang, and J. Crowcroft, "Social-aware sequential modeling of user interests: A deep learning approach," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 11, pp. 2200–2212, 2019.

[25] D. Ding, M. Zhang, S. Li, and et al., "Baydnn: Friend recommendation with bayesian personalized ranking deep neural network," in *Proc. CIKM*. ACM, 2017, pp. 1479–1488.

[26] D. Rafailidis and F. Crestani, "Friend recommendation in location-based social networks via deep pairwise learning," in *ASONAM*. IEEE Computer Society, 2018, pp. 421–428.

[27] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *NIPS*. MIT Press, 2001, pp. 585–591.

[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[29] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction." *Science*, vol. 290, no. 5500, pp. P.2319–2323, 2000.

[30] S. Abu-El-Haija, B. Perozzi, and R. Al-Rfou, "Learning edge representations via low-rank asymmetric projections," in *Proc. CIKM*, 2017, pp. 1787–1796.

[31] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proc. KDD*, 2019, pp. 732–740.

[32] C. Tu, X. Zeng, H. Wang, Z. Zhang, Z. Liu, M. Sun, B. Zhang, and L. Lin, "A unified framework for community detection and network representation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1051–1065, 2019.

[33] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proc. KDD*, 2014, pp. 701–710.

[34] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. KDD*, 2016, pp. 855–864.

[35] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: large-scale information network embedding," in *WWW*. ACM, 2015, pp. 1067–1077.

[36] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *KDD*. ACM, 2016, pp. 1225–12RendleFGS0934.

[37] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "User profile preserving social network embedding," in *IJCAI*. ijcai.org, 2017, pp. 3378–3384.

[38] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, "Network representation learning with rich text information," in *IJCAI*. AAAI Press, 2015, pp. 2111–2117.

[39] V. S. Dave, B. Zhang, P. Chen, and M. A. Hasan, "Neural-brane: Neural bayesian personalized ranking for attributed network embedding," *Data Science and Engineering*, vol. 4, no. 2, pp. 119–131, 2019.

[40] Z. Zhang, H. Yang, J. Bu, and et al., "ANRL: attributed network representation learning via deep neural networks," in *Proc. IJCAI*, 2018, pp. 3155–3161.

[41] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and et al., "Watch your step: Learning node embeddings via graph attention," in *Proc. NeurIPS*, 2018, pp. 9198–9208.

[42] S. Rendle, "Factorization machines," in *ICDM*. IEEE Computer Society, 2010, pp. 995–1000.

[43] D. Song, D. A. Meyer, and D. Tao, "Efficient latent link recommendation in signed networks," in *KDD*. ACM, 2015, pp. 1105–1114.

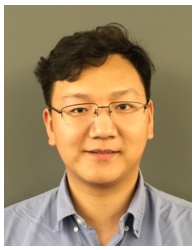
[44] P. V. Tran, "Learning to make predictions on graphs with autoencoders," in *Proc. IEEE DSAA*, 2018, pp. 237–245.

[45] S. Zhou, H. Yang, X. Wang, and et al., "PRRE: personalized relation ranking embedding for attributed networks," in *Proc. CIKM*, 2018, pp. 823–832.

[46] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *SIGIR*. ACM, 2019, pp. 165–174.

[47] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *ECML/PKDD (2)*, ser. Lecture Notes in Computer Science, vol. 6912. Springer, 2011, pp. 437–452.

Xuanzhi Zheng received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2019. She is currently a Post-Graduate at the SMILES Laboratory, School of Electronic and Information Engineering, Xi'an Jiaotong University. Hers current research interests include online dating recommendation.



Guoshuai Zhao received the B.S. degree from Heilongjiang University, Harbin, China, in 2012, the M.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2015, and the Ph.D. degree at Xi'an Jiaotong University, Xi'an, China in 2019. Now, he is an Assistant Professor at the School of Software Engineering, Xi'an Jiaotong University. He is with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, and the SMILES LAB of Xi'an Jiaotong University. He is mainly engaged in the research of social media big data analysis and recommender systems.

Li Zhu received the B.S. degree from Northwestern Polytechnical University in 1989 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University in 1995 and 2000, respectively. He is currently an Associate Professor with the School of Software, Xi'an Jiaotong University. His main research interests include multimedia processing and communication, parallel computing, and networking.

Xueming Qian (M'10) received the B.S. and M.S. degrees from Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2008. He was awarded the Microsoft Fellowship in 2006. Since 2008, he has been an Associate Professor in the School of Electronics and Information Engineering, Xi'an Jiaotong University. Now, he is an Associate Professor in the School of Electronics and Information Engineering, Xi'an Jiaotong University. He is the Director of SMILES LAB. His research interests include social media big data mining and search. He is a member of the IEEE, ACM, and Senior Member of CCF.