

What You Like, What I Am: Online Dating Recommendation via Matching Individual Preferences With Features

Xuanzhi Zheng, Guoshuai Zhao¹, Member, IEEE, Li Zhu,
Jihua Zhu², and Xueming Qian³, Member, IEEE

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. Our specific design in the model is that user preferences and user features are completely different. User features are users' personal characteristics, while user preferences indicate which kind of other users they are seeking. We aim to model user features and preferences to identify what the user is and what the 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 in USA have signed up with online dating sites to seek potential romance. About

20% of committed relationships began online. 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. Online dating sites desperately need a well-performed dating recommendation method to alleviate information overload.

The recent works about online dating mainly focus on data mining, user behavior analysis and mate preference modeling [2], [3]. For online dating recommendation task, the most of existing methods [4], [5] are based on collaborative filtering approaches. One common paradigm of collaborative filtering model is to parameterize users as embeddings, and learn the embedding parameters by reconstructing historical user interactions. However, this kind of methods lack of well-designed and complex model structure to extract features from rich user profile information. Another type of collaborative filtering methods considers the pre-existing features of a user, towards better user representations. Till now few deep learning studies which can achieve high-accuracy recommendations has been conducted to address dating recommendation. Focusing on friend recommendation, which is a task like dating recommendation, multiple methods are used to predict unknown friendship, including SVM, logistic regression, collaborative filtering [4], [6], and deep neural network [7]. They have achieved excellent performance in friend recommendation fields, but they fail to consider modeling the preference of user for dating recommendation task.

The existing dating recommendation methods face two main challenges: lacking of an appropriate way to fuse social graphs and attributes data and model comprehensive user

1. <https://www.jiayuan.com>
2. <http://cn.match.com>
3. <https://www.elitesingles.com>

- *Xuanzhi Zheng, Guoshuai Zhao, Li Zhu, and Jihua Zhu are with the School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: zhengxuanzhi@stu.xjtu.edu.cn, {guoshuai.zhao, zhujh}@xjtu.edu.cn, zhuli@mail.xjtu.edu.cn.*
- *Xueming Qian is with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, Xi'an, Shaanxi 710049, China, and also with SMILES LAB, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: qianxm@mail.xjtu.edu.cn.*

Manuscript received 23 Nov. 2020; revised 31 Oct. 2021; accepted 29 Jan. 2022. Date of publication 7 Feb. 2022; date of current version 3 Apr. 2023.

This work was supported in part by the National Natural Science Foundation of China under Grants 61902309, 61701391, and 61772407, in part by ShaanXi Provincial Natural Science Foundation under Grant 2018JM6092, in part by the Fundamental Research Funds for the Central Universities, China under Grant xxj022019003, in part by China Postdoctoral Science Foundation under Grant 2020M683496, in part by National Postdoctoral Innovative Talents Support Program, China under Grant BX20190273, in part by the Humanities and Social Sciences Foundation of Ministry of Education, China under Grant 16XJAZH003, and in part by the Science and Technology Program of Xi'an, China under Grant 21RGZN0017.

(Corresponding authors: Guoshuai Zhao and Li Zhu.)

Recommended for acceptance by K.S. Candan.

Digital Object Identifier no. 10.1109/TKDE.2022.3148485

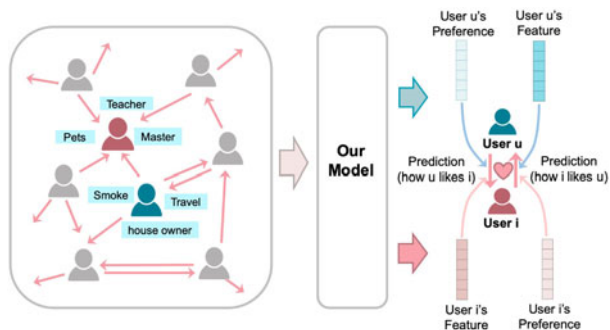


Fig. 1. An illustrative example of MIPF’s working principle. User’s feature and preference embeddings are computed with the social networks and user attributed information.

preferences. User social graphs are composed of users as nodes and interactions between users as edges. Attributes data is user profiles to introduce the basic information of users. These two different modalities are consistent and complementary. Fusing them is beneficial for model to achieve high-accuracy performance. Furthermore, existing dating recommendation methods make recommendations based on user features. They are representations of the users’ personal characteristics extracted from the user attribute information. Similar users are more likely to be recommended. However, in the real online dating recommendation scenario, users have strong preferences. The user preferences for choosing dating partners is different from their own features. Explicit preferences are the common characteristics of those users who are paid more attention to. It’s easier to be extracted compare with implicit preferences. Implicit preferences represent the common characteristics of the community members in social graph which the user is one of members. It’s also the high-order topological information extracted from the social graph.

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. An illustrative example is shown in Fig. 1. We mine explicit and implicit preferences from low-order and high-order topological information of social graphs. 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 high-order topological information of social graphs. We jointly optimize user features and preferences by combining the loss functions of graph representation task and dating recommendation task. We conduct experiments and compare MIPF with several state-of-art baseline methods on datasets collected from Shiji-JiaYuan 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 specific design in the model is that user preferences and user features are completely different. User features are users’ personal characteristics, while user preferences indicate which kind of other users they are seeking. Then, 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. Explicit preferences are captured from user interactions on online dating, while implicit preferences are mined from high-order structural social graphs.
- We jointly optimized user features and preferences by combining the loss functions of graph representation task and dating recommendation task. Experiment results demonstrate that 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

Previous studies have designed and investigated different methods for mate preference modeling and dating recommendation about online dating. Hitsch *et al.* [5] studied predicting stable online dating matches using data on user attributes and interactions. They estimated mate preferences and used the Gale-Shapley algorithm to approximate the observed online matching patterns. Su *et al.* [2] 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. 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.* [6] 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.* [4] 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 [8] 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 introduced 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.* [9] designed a two-side matching framework for online dating recommendation. It contained a Latent Dirichlet Allocation [10] 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 [11]. 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.* [12] employed a decision tree classifier to make the predictions of the co-authorship likelihood. Gong *et al.* [13] predicted new and missing links in Google+ using support vector machine.

Fitting methods [14] regard the friend relationships between users as a rating matrix, and predict the likelihood of unknown friendships through matrix factorization [15] 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.* [16] 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.* [17] 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.* [18] proposed a unified matrix-factorized framework with considering four social network factors to predict user-service rating. He *et al.* [19] 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. G. Zhao *et al.* [20] propose a context-aware model which makes full use of the users' context sensitive information through a score-ranking matrix factorization framework.

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 [21]. 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.* [22] and Rendle *et al.* [23] 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], and [26] introduced DNN into the recommendation field,

and it has proved to be effective. Ding *et al.* [7] 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.* [27] 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 [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] 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.* [32] proposed a novel approach for learning latent social representations. It used random walks to generate node sequences from graphs, then treated them as sentences through Skip-Gram. Node2vec [33] 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 [34] 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.* [35] 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 [36] that takes both attribution and relational information into account have been proposed. Zhang *et al.* [37] 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 social

TABLE 1
Symbols and Their Descriptions

Symbol	Description
G	A user social graph with users and interaction records
V	User set
E	A set of interaction records between users
X	Matrix of user attribute information
A	User adjacency matrix
f_i	Feature embedding of user i
p_i	Preference embedding of user i
s_{ui}	Prediction score of recommending user i to user u
D	Node co-occurrence matrix
P_u	Positive users for user u
N_u	Negative users for user u
K_u	Unobserved users for user u
d	Dimension of user feature and preference embedding
α	Weighting coefficient in loss function
k	Number of negative samples in training process
\circ	Hadamard product
$\ \cdot \ _1$	Sum of the matrix entities

graphs can be naturally represented as a homogeneous attributed graph $G = (V, E, X)$, 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 $i \in V$ in two low-dimensional vectors $f_i \in \mathbb{R}^d$, $p_i \in \mathbb{R}^d$ as user feature and preference embeddings. d is the dimension of the learned embeddings. Symbols utilized in this paper and their descriptions are given in Table 1.

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. 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 Fig. 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 dating partner i to user u , we need to 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 :

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

where the σ denotes sigmoid activation function. 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 users' feature and preference embeddings from attribute information. Fig. 3 illustrates the structure of node attribute encoder. The attribute information of user u is 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

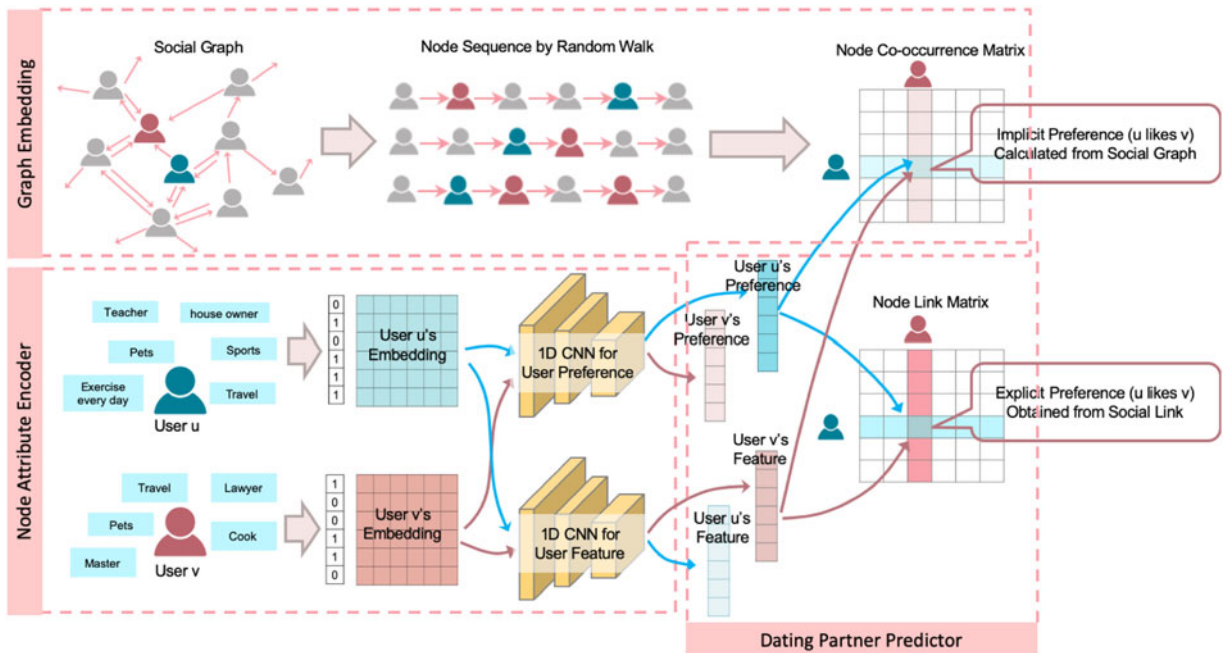


Fig. 2. The overview of our proposed model MIPF.

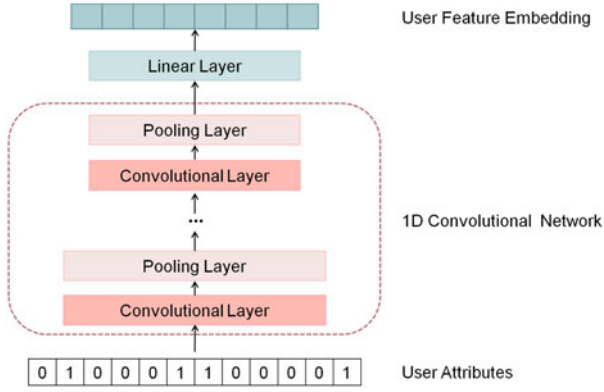


Fig. 3. The detail of node attribute encoder.

recognition, 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 verify its ability to extract features from non-sequential onehot input. Two 1D CNNs are introduced to enhance the learning of user feature and preference embeddings respectively. They have the same structure but don'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:

$$c = \tanh(W * x_u + bias), \quad (2)$$

where the W and bias denote the parameters of CNN filters. \tanh is used as the non-linear activation function. 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:

$$h = \text{pool}(c). \quad (3)$$

In order to enhance the ability of node attribute encode to capture the high-order 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:

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

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

$$f_u = \text{ReLU}(W' * h^k + bias'). \quad (5)$$

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 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:

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

We define that the indicator function $\mathbb{1}[\cdot]$ 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:

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

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 $\mathbb{1}[(u, i) \in E]$ can be replaced by simulated random walk statistic matrix D . The graph likelihood can be formulated as:

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

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 [38] 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:

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

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 edge in social graphs and its low-order topological information can be regarded as the user explicit preferences, while the potential structural information hidden in the social graphs 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, $N_u \cup K_u = \{i | i \in V, (u, i) \notin E\}$. 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 $M \in \{-1, 0, 1\}^{|V| \times |V|}$ 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 :

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

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 :

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

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

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

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 collected from the ShiJiJiaYuan website (<https://www.jiayuan.com/>), one of the most famous online dating sites in China. Compared with other online dating websites, it has a large number of users with comprehensive user profiles. Moreover, the interaction records between users are showed on the website, which is the basic of constructing user social graphs.

TABLE 2

The Proportion of Three Categories of Interaction Records in SDMCD

Categories	Recommendation	Click	Message
Proportion	97.29%	2.14%	0.57%

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.^{4,5} It contains anonymized profiles and heterosexual dating interaction records during a period of 3 months. It contains 344,522 male users and 203,843 female users. Each user owns 34 personal attributes, which are showed in detail in Table 3. 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. These records can be divided into three categories: recommendation, click, and message. Recommendation records represent that a user was recommended to another user. Click records indicate that a user clicked and browsed the homepages of a recommended user. Message records indicate that a user messaged to other users after browsing their homepages. 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 one record between A and C belonging to recommendation category. The proportion of these each category to the total interaction records is shown in Table 2.

We compute the conversion probability between these three categories. The results are shown in the Fig. 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 users with click records and the message records can be merged as correct recommended user pairs.

Considering the training time and the runtime memory requirements of our method and baselines, we try to generate a small dataset for experiments. One way is to sample users completely randomly, and then retain the records whose both vertices are sampled. However, the sampled users are sparsely distributed on graphs, and only a small proportion of records are retained. We hope that the proportion of users and records in the small dataset is close to that of the original dataset. Therefore, we randomly select a user whose interaction records are more than 20 as the seed user, and then add

4. <https://github.com/Zhengbaibai/ShiJiJiaYuan-Competition-Dataset>

5. <https://cosx.org/2011/03/1st-data-mining-competition-for-college-students/>

TABLE 3
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 4
Self-Description Features in ShiJiJiaYuan Crawled Dataset

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

the users which has interaction records with the seed user (also the first-hop neighbors of the seed user) to the small dataset. We iteratively add the neighbors of the users in the small dataset, until the fifth-hop neighbors of the seed user are added. The number of users reaches 15,669 and 180,816 interaction records are 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 interaction records. The dataset has 3,146 male users accounting for 28.04% and 7,923 female users. User profiles have 64 fields, which are listed in Table 4. Some of fields are required fields such as age, sex, marital status, and so on. Other fields like jobs, fitness frequency, education level and family members can be omitted. The user relationships in SCD are the records of following actions in the ShiJiJiaYuan site. Users can follow other users and track their social dynamics.

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 user popularity in dating market. In Fig. 5a, 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 Fig. 5b, 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 decrease their attractiveness. In addition, we find that there is difference between men and women's preferences in some specific attributes such as monthly income, showed in

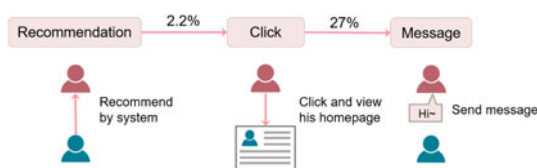


Fig. 4. The probability of having subsequent actions.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on July 09, 2023 at 18:09:21 UTC from IEEE Xplore. Restrictions apply.

Fig. 5c. 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 graph embedding algorithms. Friends recommendation methods contain fitting approaches, FM [39], and ranking approaches, BPRMF [40] and BayDNN [7]. Four network embedding methods, LoNGAE [41], PRRE [42], NGCF [43], Graph Attention Model [38], are also trained to generate recommendations by computing the cosine similarities between users.

- 1) *FM* [39]: A basic factorization machine for prediction problem, which can mimic most factorization models just by feature engineering. Menon and Elkan *et al.* [44] proposed to solve the link prediction problem using factorization machines and regarding positive links as 1 and the others as 0.
- 2) *BPRMF* [23]: 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 in the field of personalized ranking.
- 3) *LoNGAE* [41]: 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* [42]: 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* [7]: BayDNN combines Bayesian Personalized Ranking and Deep Neural Network for friend

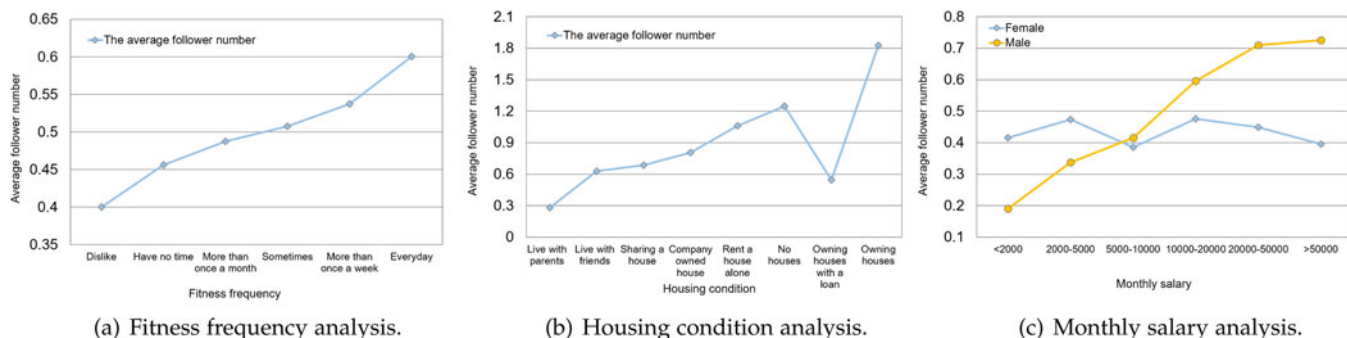


Fig. 5. Average user follower number with different attributes on SCD.

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* [43]: 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.
- 7) *Graph Attention Model* [38]: 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.

4.3 Implementation

The user profiles in the two datasets contains various fields, which are comprehensive description of users. However, some fields are omitted by most of users. Some fields are free texts, which are 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 onehot processing to convert the Chinese categorical options and numeric data into vectors containing only 0 and 1.

As user interaction records, we use negative sampling techniques. In SDMCD, the users who are clicked or sent messages by other users can be regarded as positive samples. We randomly sample K users which are not interacted with this user or only recommended to this user as negative samples. In SCD, we regard the users which are followed as positive samples. 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 training set, and 20% positive samples for 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 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 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 default settings in their 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 hyper-parameters according to the paper of BayDNN. The dimension of user embeddings 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 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 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 5. 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 AUC. Although Graph Attention Model introduces attention machine in simulation random walk to capture the higher-order proximity information of social graphs with trainable

TABLE 5
Performance Comparison on AUC, AP and P@n

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%

hyperparameters, they have a disadvantage that they do not take side information into account. Our approach designs a node attribute encoder module and a prediction loss function to extract the latent information in user attributes. Thus, our approach achieves better performance. Compared with BayDNN, our approach not only takes advantage of 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, 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 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 feature and preference embeddings, we compare original MIPF method with MIPF-SE. To study the effect of the loss function and the node attribute encoder module, we compare MIPF with different loss function and MIPF without the node attribute encoder module.

In summary, the experimental configurations are detailed below:

- 1) *MIPF*: The original MIPF model. The loss function is the weighted sum of perdition loss and NLGL loss.
- 2) *MIPF-NAE*: MIPF model without user attribute input and the node attribute encoder module.
- 3) *MIPF-MLP*: MIPF model with MLP layers instead of the node attribute encoder module.
- 4) *MIPF-SE*: MIPF model with single user embedding.
- 5) *MIPF-BPR*: MIPF model with BPR loss function.
- 6) *MIPF-PL*: MIPF model with perdition loss function.
- 7) *MIPF-NLGL*: MIPF model with NLGL loss function.
- 8) *MIPF-NLGL-BPR*: MIPF model with BPR and NLGL loss function, the sum weight of two loss functions is 1.

In order to make a fair comparison, the dimension 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 6. The performance comparison of MIPF-SE,

TABLE 6
Ablation Study Results on AUC, AP and P@n

Methods	SCD					SDMCD				
	AUC	AP	@P1	@P5	@P10	AUC	AP	@P1	@P5	@P10
MIPF	0.831626	0.859338	0.558602	0.767649	0.875600	0.852843	0.859331	0.467030	0.809847	0.905152
MIPF-SE	0.801323	0.817145	0.526388	0.744688	0.866690	0.819290	0.789203	0.308199	0.778753	0.900617
MIPF-NAE	0.804417	0.824549	0.530500	0.759424	0.870802	0.819113	0.837645	0.461287	0.785311	0.903426
MIPF-MLP	0.829873	0.855223	0.525195	0.779807	0.887252	0.841575	0.853395	0.456586	0.803681	0.909982
MIPF-PL	0.637188	0.613786	0.315970	0.471899	0.665867	0.668893	0.672681	0.236288	0.569807	0.730399
MIPF-BPR	0.714743	0.733363	0.378684	0.622001	0.778958	0.741513	0.761121	0.263108	0.629450	0.792163
MIPF-NLGL	0.823927	0.850759	0.542018	0.761526	0.870682	0.844659	0.848977	0.441923	0.804496	0.906849
MIPF-NLGL-BPR	0.829312	0.849029	0.560573	0.750526	0.873994	0.847470	0.846857	0.439101	0.807106	0.907501

TABLE 7
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

MIPF proves the effectiveness of distinguishing feature and preference embeddings in dating recommendation. Although the performance degradation of MIPF-SE can be partly attributed to the decrease of learnable parameters from $O(2 \times |V| \times d)$ to $O(|V| \times d)$, it can be seen from the experiments on the parameter d that MIPF still has better performance under the same learnable parameter number. Focusing on MIPF-NAE, we find that the performance is degraded on both datasets, which demonstrates the effectiveness of the node attribute encoder module. To verify the effectiveness of CNN in the node attribute encoder module, we compared the MIPF method and the one with multi-layer perceptron (MLP). The hidden unit number of MLP is 256,128,128 for SDMCD and 128,64,8 for SCD. We find that the performance of MIPF and MIPF-MLP is close. The AUC and AP values of our MIPF are a bit higher than MIPF-MLP on both datasets. We can conclude that both using 1D CNN and using MLP could learn a good user representation for dating recommendation, and the overall performance of using 1D CNN is slightly better than using MLP. Comparing MIPF with different loss function, we find that NLGL loss function consistently outperforms when these three loss functions are used alone. This is because the other two loss functions cannot learn the high-order topology information from user graphs. When compared the MIPF-NLGL-BPR with MIPF, their performance is very close, it proves that using the weighted sum of NLGL loss and BPR loss as total loss function is also a feasible way.

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 Table 7. 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 0.000001 on SCD and is 0.000001 on SDMCD. At this time, 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 graphs which is the explicit preferences. The implicit preference hidden in high-order information of the social graphs 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. The experiments results about d are shown in Tables 8 and 9. 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. 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 on two

TABLE 8
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 9
Discussion on the Parameter d on SDMCD

d	AUC	AP	@P1	@P5	@P10
32	0.835077	0.841435	0.466638	0.791151	0.887533
64	0.852153	0.857100	0.462168	0.810336	0.905772
128	0.852843	0.859331	0.467030	0.809847	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

TABLE 10
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

dataset 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 complex information.

4.4.4 Discussion on the Hyperparameter K

To investigate how parameter K affects the performance, we vary number of negative samples for each positive sample in the training process. In particular, we change the K value in the range of $\{1, 2, 4, 8, 16, 32, 64\}$. Table 10 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 Interaction Records

We want to explore the performance of MIPF on different sparseness data. 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 Fig. 6, 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 overmuch social connections will bring noise in the process of graph embedding.

5 CONCLUSION

This paper presents an online dating recommendation model MIPF by matching individual preferences with features. User attributes and social graphs are fused to conduct user feature and preference embeddings. The user preference embedding contains explicit preference as the low-order topological information, and implicit preference which is the high-order topology information in social graphs. 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 texts such as

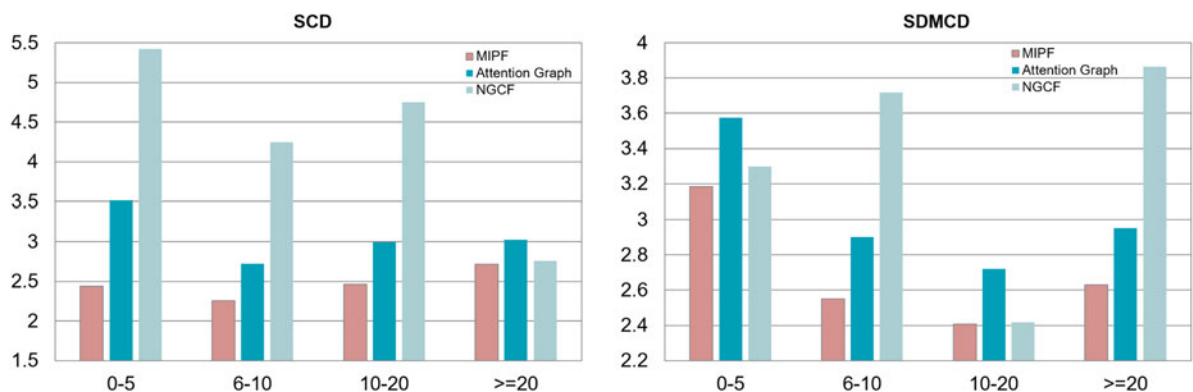


Fig. 6. The Performance of MIPF and two comparison methods on user groups with different number of interaction records on SCD and SDMCD. The y-axis is the ranks of positive users in candidate user lists. The lower average ranking denotes the better recommendation for users.

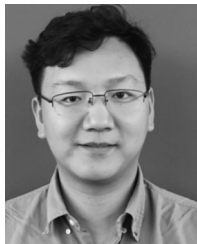
self-introduction of users. With the help of the latest Natural Language Processing technology, extracting the information in free texts can make the model achieve better performance. Secondly, the current graph embedding methods have relatively high requirements for running memories when extracting high-order 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. Anal. Mining*, vol. 6, no. 1, pp. 32:1–32:16, 2016.
- [2] X. Su and H. Hu, "Gender-specific preference in online dating," *EPJ Data Sci.*, vol. 8, no. 1, pp. 12:1–12:22, 2019.
- [3] A. Kleinerman, A. Rosenfeld, and S. Kraus, "Providing explanations for recommendations in reciprocal environments," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 22–30.
- [4] 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 *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2011, pp. 2199–2204.
- [5] G. J. Hitsch, A. Hortaçsu, and D. Arieli, "Matching and sorting in online dating," *Amer. Econ. Rev.*, vol. 100, no. 1, pp. 130–63, 2010.
- [6] 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, Mar./Apr. 2014.
- [7] D. Ding et al., "BayDNN: Friend recommendation with bayesian personalized ranking deep neural network," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1479–1488.
- [8] L. Li and T. Li, "MEET: A generalized framework for reciprocal recommender systems," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 35–44.
- [9] K. Tu et al., "Online dating recommendations: Matching markets and learning preferences," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 787–792.
- [10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [11] 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. Manage. Inf. Syst.*, vol. 9, no. 1, pp. 1:1–1:26, 2018.
- [12] N. Benchettara, R. Kanawati, and C. Rouveirol, "A supervised machine learning link prediction approach for academic collaboration recommendation," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 253–256.
- [13] N. Z. Gong et al., "Joint link prediction and attribute inference using a social-attribute network," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 27:1–27:20, 2014.
- [14] 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, Jul. 2014.
- [15] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [16] Y. Yang, Y. Xu, E. Wang, J. Han, and Z. Yu, "Improving existing collaborative filtering recommendations via serendipity-based algorithm," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1888–1900, Jul. 2018.
- [17] 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, Apr. 2017.
- [18] 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, Sep. 2016.
- [19] 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," *Concurrency Computation Pract. Experience*, vol. 29, no. 6, 2017, Art. no. e3924.
- [20] G. Zhao, Z. Liu, Y. Chao, and X. Qian, "CAPER: Context-aware personalized emoji recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 9, 2021, pp. 3160–3172, doi: [10.1109/TKDE.2020.2966971](https://doi.org/10.1109/TKDE.2020.2966971).
- [21] H. Tang, G. Zhao, Y. Wu, and X. Qian, "Multisample-based contrastive loss for top-k recommendation," *IEEE Trans. Multimedia*, early access, Nov. 15, 2021, doi: [10.1109/TMM.2021.3126146](https://doi.org/10.1109/TMM.2021.3126146).
- [22] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 173–182.
- [23] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.
- [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, Nov. 2019.
- [25] J. Hao, Y. Dun, G. Zhao, Y. Wu, and X. Qian, "Annular-graph attention model for personalized sequential recommendation," *IEEE Trans. Multimedia*, early access, Jul. 14, 2021, doi: [10.1109/TMM.2021.3097186](https://doi.org/10.1109/TMM.2021.3097186).
- [26] Y. Wu, K. Li, G. Zhao, and X. QIAN, "Personalized long-and short-term preference learning for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Jun. 15, 2020, doi: [10.1109/TKDE.2020.3002531](https://doi.org/10.1109/TKDE.2020.3002531).
- [27] D. Rafailidis and F. Crestani, "Friend recommendation in location-based social networks via deep pairwise learning," in *IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, 2018, pp. 421–428.
- [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-Hajja, B. Perozzi, and R. Al-Rfou, "Learning edge representations via low-rank asymmetric projections," in *Proc. Int. Conf. Knowl. Manage.*, 2017, pp. 1787–1796.
- [31] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 732–740.
- [32] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.
- [33] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.
- [34] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.
- [35] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1225–1234.
- [36] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "User profile preserving social network embedding," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3378–3384.
- [37] Z. Zhang et al., "ANRL: Attributed network representation learning via deep neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3155–3161.
- [38] S. Abu-El-Hajja et al., "Watch your step: Learning node embeddings via graph attention," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9198–9208.
- [39] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.
- [40] D. Song, D. A. Meyer, and D. Tao, "Efficient latent link recommendation in signed networks," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1105–1114.
- [41] P. V. Tran, "Learning to make predictions on graphs with autoencoders," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal.*, 2018, pp. 237–245.
- [42] S. Zhou et al., "PRRE: Personalized relation ranking embedding for attributed networks," in *Proc. Pro. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 823–832.
- [43] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.
- [44] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2011, pp. 437–452.



Quanzhi Zheng received the BS degree in 2019 from Xi'an Jiaotong University, Xi'an, China, where she is currently working toward the postgraduation degree with the SMILES Laboratory, School of Software Engineering. Her current research interests include recommender systems and social media big data analysis.



Guoshuai Zhao (Member, IEEE) received the BS degree from Heilongjiang University, Harbin, China, in 2012, and the MS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively. He was an intern with Social Computing Group, Microsoft Research Asia from January 2017 to July 2017, and was a visiting scholar with Northeastern University from October 2017 to October 2018, and with MIT from June 2019 to December 2019. He is currently an associate professor with Xi'an Jiaotong University. His

research interests include social media big data analysis, recommender systems, and natural language generation.



Li Zhu received the BS degree from Northwestern Polytechnical University in 1989, and the MS and PhD degrees from Xi'an Jiaotong University in 1995 and 2000, respectively. He is currently a full professor with the School of Software, Xi'an Jiaotong University. His main research interests include multimedia processing and communication, parallel computing, and networking.



Jihua Zhu received the BS degree in automation from Central South University, China, in 2004, and the PhD degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, in 2011. He is an associate professor with the School of Software Engineering, Xi'an Jiaotong University. His research interests include mobile robot, computer vision, and machine learning.



Xueming Qian (Member, IEEE) received the BS and MS degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the PhD degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2008. He is currently a full professor with Xi'an Jiaotong University. He is also the director of SMILES Laboratory. His research interests include social media big data mining and search. He is a member of ACM, and a senior member of CCF.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.