IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2020, accepted March 18, 2020, date of publication March 30, 2020, date of current version April 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984340

An Efficient Adaptive Attention Neural Network for Social Recommendation

MUNAN LI, KENJI TEI, AND YOSHIAKI FUKAZAWA, (Member, IEEE)
Waseda University, Tokyo 169-8050, Japan
Corresponding author: Munan Li (limunan@asagi.waseda.jp)

ABSTRACT Traditional recommendation algorithms based on collaborative filtering suffer from a data sparsity problem. The emergence of online social network has enriched the user's information, realizing a new way to solve recommendation tasks. Social-aware recommendation algorithms can effectively alleviate the data sparsity problem and improve the performance of recommendation systems. Despite the success of these algorithms, they have some common limitations. Most algorithms assume that social networks are homogeneous, with similar preferences among connected users. However, users may only share similar preferences in some aspects. Besides, different friends affect the user's preference in different levels. And this influence of friends on users' preference should be adaptive. Even close friends may have different influences in different decision-making processes. For example, a user may trust a friend in "travel" but distrust this friend in "music" because this friend had more travel experiences. Motivated by the above limitations, we designed a neural network model called adaptive attention neural network for social recommendation (ANSR) to study the interaction between a user and his or her social friends as well as infer the complex influence of the user's social relationships on the user's preferences. By utilizing the co-attention mechanism, we can not only extract the user's special attention to certain aspects of their friends but also determine the adaptive influences of different friends on the user. When the user interacts with different items, different attention weights will be assigned to the user and his or her friends, respectively. In addition, we also utilize network embedding to learn more efficient features of each user and incorporate these features into the ANSR to enhance the recommendation results. Moreover, we also conduct extensive experiments on four different real-world datasets and demonstrate that our proposed method performs better on all datasets compared with the state-of-the-art baseline methods.

INDEX TERMS Recommender systems, neural networks, attention, social influence.

I. INTRODUCTION

A personalized recommendation system can effectively alleviate the problem of information overload caused by the explosive growth of the Internet. On the one hand, it can enable users to discover items of interest from a large quantity of information. On the other hand, it can help information-providers present information to interested users to achieve profitability. A traditional recommendation system based on collaborative filtering infers the interest of each user based on the historical information of the user's project interaction. Matrix Factorization (MF) [25] is the most common method, which decomposes the user-item interaction matrix into two low-dimensional vectors to learn the user's preference and item characteristic. For some large websites such as the Amazon website, the number of user-rated items is quite small, which leads to an extremely sparse user-item rating matrix. This affects the performance of the recommendation system.

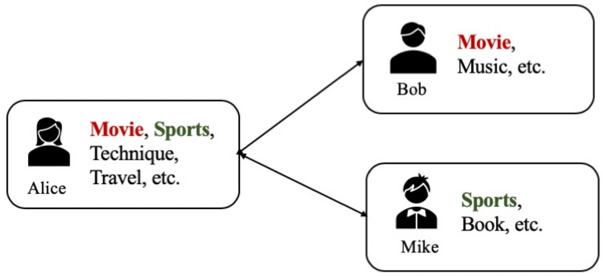
Luckily, the emergence of online social platforms has enriched the user's information. On these platforms, the user is able to not only rate or click items he or she is interested in, but also can interact with his or her friends. The user's preferences are determined by his or her history of rating or clicking information, as well as the influence of friends. Therefore, many studies exploit social relations for collaborative filtering to address the data sparsity problem. In studies of social-aware recommendation systems, researchers have found that the user's preference is similar to or influenced by his or her friends. On the basis of this conclusion, many social-aware recommendation methods [2], [6], [7], [26]–[28] integrated the users' social connections into the

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqiang Wang.

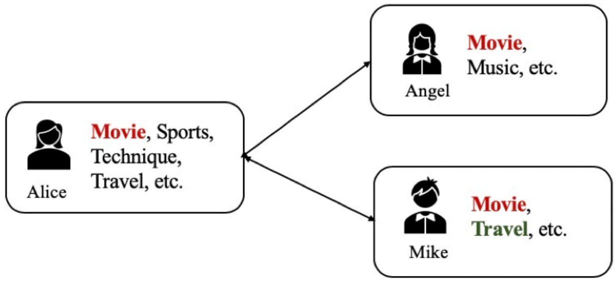
collaborative filtering algorithms and utilized the social information to enrich the user's features to mitigate the problem of data sparsity. For example, Jamali and Ester [2] proposed a trust propagation method called SocialMF, which built on matrix factorization to solve social recommendation tasks. These previous studies such as SocialMF all assumed that the user was equally influenced by each friend and the influence values are assigned with the same weight. Such a uniform weight assignment strategy as uniform weight can only roughly estimate the influence of social relations on users and cannot accurately measure how does the social relations affect the user's preferences. To overcome this limitation, some studies [3], [5], [17], [19] tried to calculate the influence value given by each friend to the user while leveraging social relations on recommendations. For example, Chaney *et al.* [3] integrated social relations into Poisson factorization (PF) and proposed a probabilistic model in which PF captures the different influence values of friends for each user.

Although the above algorithms have achieved great success, they have some common limitations. First, most studies assume that social networks are homogeneous and share similar preferences among a linked user community. However, a user's concern depends on the friend and the context. As shown in Fig.1(a), the users have different concerns about different friends, and only share similar preferences in some aspects. The user may be interested in one aspect of a friend's preference but focus on another aspect of a different friend. For example, a sports fan may be influenced by friends from different communities when choosing a new music album. Therefore, how to capture similar parts of the user's preferences with different friends must be considered when utilizing social relations for recommendations. Second, most studies assume a uniform influence weight between users and their friends. However, the relationship types between the user and friends in online social networks vary, for examples, close friends and common acquaintances. The user's trust depends on the type of friends. On a social platform, the user usually clicks on news or items recommended by a friend who communicates frequently with the user while the user tends to consider a friend who does not often communicate with him or her to be relatively untrustworthy. The influence of friends on the user's decision-making on social platforms is a more complex process that should be adaptive. Even close friends will have different influences in different decision-making processes. For example, as shown in Fig.1(b), the user may trust a friend when seeking "travel" advice, but not when seeking "music" advice. This is because this user's friend has more experience of travel than with music. In summary, most studies focus on how to utilize the user's social connections in the recommendation process to enhance the recommendation results. Typically, social influences between users are set equally based on equal social relationships, or other predefined strategies based on data mining.

Motivated by the above limitations, herein, we focus on the main problem for social recommendations – how is the



(a)



(b)

FIGURE 1. **Illustration of the trust varies among categories, (a) shows that users only share similar preferences in some aspects and (b) shows that users trust different friends in different trust levels.**

user's preference affected by the user's social connections. Unlike the above-mentioned equal setting or other predefined strategies, the interaction between the user and social friends is learned through a neural network to infer the complex influence of social relationships on the user's preferences. This paper proposed an efficient adaptive attention neural network for social recommendation (ANSR) model, which is a novel architecture based on a co-attention neural network. The model divides the factors that determine the user's preferences into two: item-based and social-based preference. The former is based on the user's past interactions with the item, while the latter is determined by the complex effect between the user and the user's social relations. In particular, we design a module that leverages the co-attention neural network to learn the social effect between the user and his or her friends. The co-attention module is a two-way perception neural network. On the one hand, it captures the user's specific attention on an aspect of his or her friend and outputs the representation of the user's specific preference. On the other hand, it captures the degree of influence of different friends on the user. Consequently, the output is the influence representation of the user's friends. Then, these two representations are joined to constitute the representation of a social-based preference; along with the representation of an item-based preference to alleviate the data sparsity problem. In addition, this paper is an expansion of the paper that was presented at the International Conference of Web Intelligence [1]. New datasets were used in the experiments. The model proposed in this paper was improvement over the previous version. Besides, new experiments were conducted to analyze the validation of the model.

The contributions of this work are summarized as follows:

1) The user's specific preference and social influence preferences are distinguished from the viewpoint of social effects. Investigating these two effects in recommendation systems enables the model to capture more information and improve the recommendation performance.
2) A model based on a co-attention neural network is designed to learn the social effect between the user and friends. It captures the user's specific attention on an aspect of friends and outputs the user's social specific similarity. On the other hand, it captures the degree of influence of different friends. Consequently, it outputs the social influence representation of the user's friends.
3) Moreover, ANSR is extended by using the network embedding. The network embedding algorithm is utilized to learn more efficient features of the user and enhance the recommendation effect.
4) We conducted many experiments on four real-world data sets to demonstrate the superiority of our proposed method over the state-of-the-art methods and the effectiveness of the key components of ANSR.

II. RELATED WORKS

A. SOCIAL RECOMMENDATIONS

In recent years, utilizing social connections to improve the performance of the recommendation system has been very successful. Researchers have found that users and their social connections share similar preferences. According to this theory, some researchers have tried to integrate the user's social connections into the matrix factorization-based method to enrich the user's features and alleviate the data sparsity problem. For example, an matrix factorization method called SocialMF [2] was proposed by Jamali and Este to leverage the user's social connections for the recommendation system. In SocialMF, the authors believed that the user's latent preference depends on his or her friends and sum all the preference features of the user's friends to obtain the user's new feature. Unique to factorization-based methods, Guo *et al.* [18] utilized social connections to extend SVD++ to learn both the user preferences and the social influence. In the previous work, all friends had the same influence on the user. However, in a real scenario, the influence of different friends will vary. For example, the user will trust close friends more than acquaintances.

To overcome this limitation, other works have tried to measure the different influences between different users when utilizing social connections to solve recommendation tasks. For example, Fazeli *et al.* [5] proposed a model based on SoicalMF and utilized implicit trust scores to enhance the performance of model-based recommendation systems. The trust scores between users were computed by the trust metrics in the literature collected by the user. Ye *et al.* [6] proposed a method involving assigning weights to different friends to improve the accuracy of recommendation systems. These influence weights were calculated by using the similarity of the user's past interaction with items and the similarity of the user's social connections. Wang *et al.* [23] proposed an EM-based model that built on the BPR model to leverage the user's social information. This paper distinguishes two user social types, strong and weak, and tries to assign an higher influence weight to strong type social friends. Zhang *et al.* [24] designed a collaborative user network embedding (CUME) model to solve the recommendation tasks. The CUME model can generate top-k semantic friends by learning from the user's rating history and social connections. Then, the CUME model tends to assign a higher influence weight to the most similar friends. Nevertheless, none of the above-mentioned work considered that the influence weight between the users and the users' friend will change depending on the process in which the user interacted with different items. Our method aims to solve this problem.

B. DEEP LEARNING-BASED RECOMMENDATIONS

In recent years, models based on deep learning have been widely used in computer vision, image recognition, natural language processing, and other research fields. Utilizing deep learning to solve recommendation tasks has also attracted much attention. This is because a recommendation model based on a neural network can effectively learn the linear and nonlinear interaction between the user and the item, and automatically extract the effective feature representation from the input information. Consequently, the limitations of the a traditional recommendation system can be overcome.

To improve the accuracy of recommendation systems, many researchers utilize deep learning to learn a better representation from rich input data sources, such as the item's content, commands from the user, and visual information. For example, a novel convolution matrix factorization (ConvMF) model was proposed by Kim *et al.* [8]. The ConvMF model integrates the convolution neural network (CNN) into a matrix factorization model to learn a better item representation and capture the contextual features from text data to improve the recommendation tasks. Besides the content information, other sources can be also be used in a deep learning-based model. For example, a joint representation learning (JRL) framework which based on deep learning architecture was proposed by Zhang *et al.* [9] to learn a better representation of the user and item. The model can jointly learn the representations of the user and item by using different types of information sources (review text, product image, and rating). Hidasi *et al.* [10] proposed a session-based recommendation model that utilizes a recurrent neural network to learn from the sequential purchase information for real-world e-commerce websites to predict what the user will buy next. In particular, a general framework called neural network-based collaborative filtering (NCF) was proposed by He *et al.* [11] to learn the linear and nonlinear interaction between the user and item. It has been proved that the NCF model has outperformed the factorization-based method.

The attention mechanism is a useful tool for deep learning. It can be combined with neural network models such as MLP, CNN, and RNN to improve model performance. The advantage of applying the attention mechanism to a deep learning-based model is that it can filter out useless information and capture the most relevant features by calculating an attention score to obtain a better representation. For example, an attention neural network-based recommendation model was proposed by Cheng *et al.* [12] to improve the accuracy of the prediction model. First, the user's latent features and item's latent features are learned through a topic-based model from the review text. Then, they utilized an attention network to learn the attention score of the user on each aspect of the item. By learning the user's attention scores for different aspects of the item, the model can accurately know the user are more interested in which aspect of the item. In the recommendation system, the attention mechanism has been proved to be an effective mechanism with better performance than other neural network technologies. Our method also utilizes an attention mechanism to extract useful features from input information and improve the performance of social-based models.

C. NETWORK EMBEDDING

Recently, network embedding has attracted great attention as a new network analysis paradigm. Traditional topology-based network representation learning can learn the representation of nodes from the adjacency matrix directly. However, such sparse representation does not apply to machine learning methods. To overcome this limitation, network embedding becomes a new solution. Network embedding can learn the low dimensional representation of nodes in the network. In addition to effectively saving the content information of nodes, this also effectively retains the network structure information. In this work, we also utilize network embedding to learn the node representation to address the data sparsity problem.

Different methods can be adopted to map networks from the original network space to the embedding space and learn the representation of nodes. For example, DeepWalk [13] can learn the social representation of a network by a random walk. Even if there are few nodes marked on the network, it can still obtain a better result. Grover and Leskovec proposed a novel method called Node2vec and designed a second-order random walk strategy to sample neighbor nodes and maintain node neighbor formation [14]. This method utilized the skip-gram model to solve the network representation learning problem. LINE [15] considers both the local and global features of the network, and it is suitable for any size of network. SNDE [16] aims to learn the Gaussian distribution in a low-dimensional space as a representation of the nodes in the network, which simultaneously maintains the network characteristics and reflects the uncertainty of the nodes.

III. METHODOLOGIES

This section introduces the motivation and intuition of our approach in detail. First, we present the whole architecture of our proposed model. The framework of our proposed model contains four components: the embedding layer, the co-attention layer, the pooling layer, and the prediction layer. Then, we describe these four layers in detail. Finally, we describe how to optimize the variables of the proposed method.

A. GENERAL FRAMEWORK

Our proposed model aims to learn a better user representation and item representation to predict unknown ratings. The whole architecture of the proposed model is showen in Fig.2. The proposed model divides the features that determine the user's latent preferences into two: the item-based and social-based preferences. To learn these features, the model first maps the raw input of the user and the item to low-dimensional space through the embedding layer. Since there is a common assumption that the user's preferences are similar or affected by his or her friends in the social-aware recommendation system, the proposed model focuses on a key issue — how the user's social relations affect the user's latent preference. As a key part of the model, the co-attention layer is utilized to learn the features that affect the user's preferences. After that, a fusion layer is used to fuse these features into a synthetic layer. The predicted score is learned through a fully connected layer. Then, we go through the details to explain the motivations and the technique of the model.

1) EMBEDDING LAYER

The model first takes the rating matrix R and the user's social network G as input. We first consider how to get the item's embedding as the representation of the item's characteristic. The input of each item can be represented by a one-hot encoding of the item's identity number. As this one-hot encoding is a sparse vector with a high dimension, the embedding layer maps it to a dense space to get a low dimensional vector. Then, the obtained item representation can be viewed as the latent feature of the item, which is used to infer the item's attribute.

Compared to item embedding, the user's embedding is more complicated. The user's preference can be inferred from the purchase or click history. In addition, it is affected by the user's social connections. We define the representation of the user's preference as having two parts. One is the item-based preference, and the other is the social-based preference. The former representation infers the user's preference in the context of the user-item history interaction. The latter representation represents the user's social-based preference in the context of the user-user history interaction. The user's item-based preference can be obtained in the same way as the item embedding was obtained. We use the embedding operation to map the sparse one-hot vector of the user's identity to a dense representation. The obtained representation is viewed as the user's item-based representation.

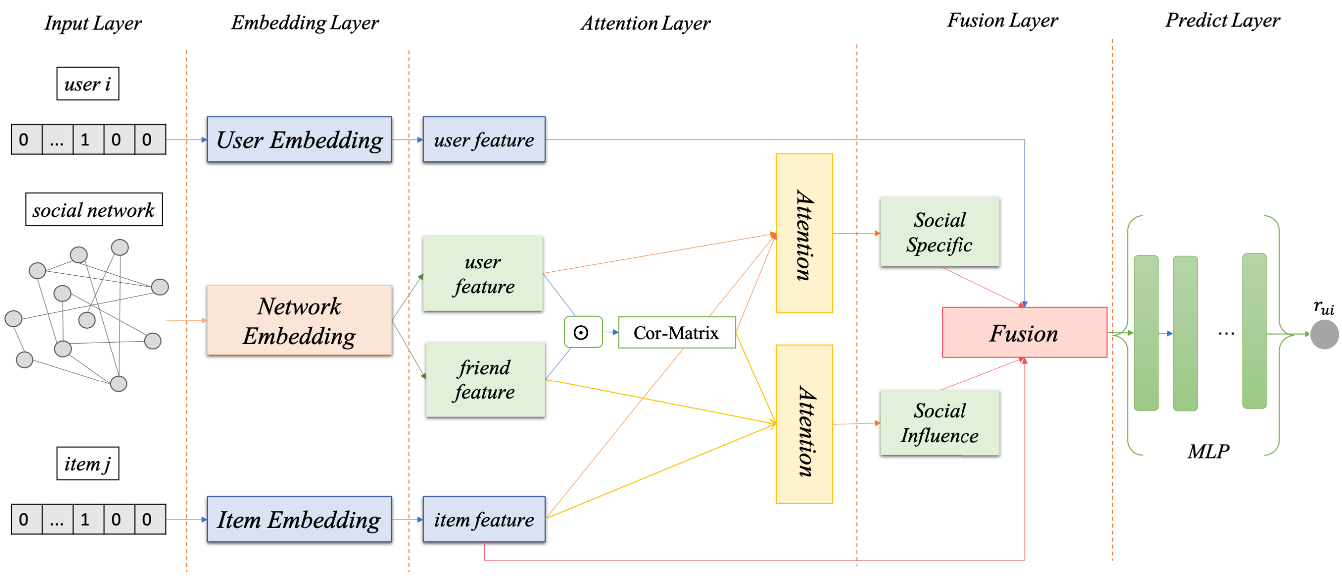


FIGURE 2. Architecture Framework.a) The model is divided into five layers: input layer, embedding layer, attention layer, fusion layer and prediction layer. b) The input layer requires the user-item interaction and users' social relations as input. c) The embedding layer embeds the users' and items' one-hot encoding into a low-dimensional space to get the embedding of the users and items. Network embedding embeds the users' social network into a low-dimensional space to get the node embedding of the users. d) The attention layer learns users' preference under the users' social effect. e) The fusion layer fuses the representations of the users and items into a synthetic layer. e) The prediction layer predicts the probability of user *u* clicking the item *i*.

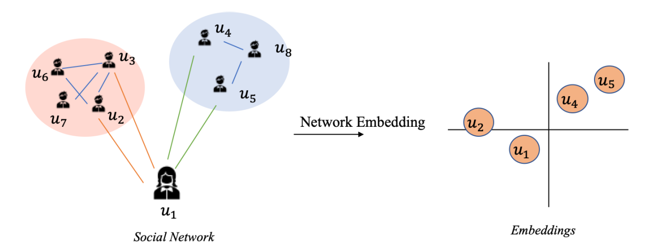


FIGURE 3. Network Embedding. Network embedding maps users' identity onto a low-dimensional space to learn the node representations while preserving the network topology and content information.

2) NETWORK EMBEDDING LAYER

Most existing works directly utilized the adjacency matrix to learn the latent feature representation of the user. Such methods do not effectively extract the latent features of nodes in a social network. As shown in Fig.3, network embedding can represent the nodes in a network as low-dimensional vector representations while preserving the network topology and content information to perform subsequent graph analysis tasks. To fully utilize the user's social network to enhance the representation of the user, we use the network embedding method to learn the low-dimensional representation of users in the social network. First, we use user relation network G as the input; the network embedding operation will output a new representation for each user. Numerous network embedding methods exist such as Node2Vec and SDNE. Here, we utilize a network embedding method (SDNE) to learn the new representation of the user. The representation learning of the user through a graph neural network can be viewed as a strategy, where the obtained features are entered into the next layer to learn the user's social-based representation.

3) ATTENTION LAYER

Adding attention mechanisms to neural network models has become a successful method to improve the performance of predictive models.Researchers have successfully used attention models to improve the model's accuracy in many research fields, such as image recognition. Actually, the attention model can be viewed as an effective feature extractor that extracts the most relevant data that is useful for prediction. Therefore, we can utilize the attention mechanism to extract more useful user features, which helps us to predict the unknown rating more accurately between the user and the item. The benefits of incorporating the attention mechanism into the proposed model are obvious. There is a special correlation between the feature vectors of the user and a friend. The user tends to share similar preferences with the user's friend. Meanwhile, the friend's preferences are affected by the user. We can use the user and friend's correlation matrix to learn about this special relevance characteristic. Through the correlation matrix, we can extract the special attention of the user regarding a certain aspect of the friend as well as the varying influences of different friends on the user. In addition, the attention mechanism can adaptively assign different attention scores to friends when the user interacts with different items. Through this attention mechanism, we can obtain adaptive feature vectors of the user to further improve prediction accuracy.

Through the network embedding of the user's social network, the representation of each user is obtained based on the network structure. Taking these representations as the input vectors of the user and his or her friends, these vectors are first used to calculate the correlation matrix for the user and his or her friends. There are different ways to calculate the correlation matrix for users and users' friends, such as the element-wise product method, and the multi-layer perceptron method. During the experiment, we found that the multi-layer perceptron method outperformed other methods. Therefore, we utilized the multi-layer perceptron method to calculate the correlation matrix. Correlation matrix C of the user i and his or her friends is learned through the following operations?

$$C_i = ReLU(F_i * W_{c1} + \widetilde{u_i} * W_{c2} + b_c)$$

where $\widetilde{u_i}$ is the matrix of user i, F_i is the matrix of user i's friends with each friend's vector f_i, and W_c and b_c are the weight matrices and bias, respectively. To match the matrix size of the user's friends, we resized the user's vector to the same dimension as the friends' matrix. ReLU is adopted as the activation function to increase the nonlinear learning ability between each layer in the model. The correlation matrix can be viewed as the preference similarity matrix between the user and his or her friends. Through the correlation matrix, we can learn useful preference features of the user. Next, the correlation matrix is put into the attention layer to learn the specific feature with regard to the social effect. To learn the social specific attention on one aspect of the user's friend, the user's features and the user's correlation matrix with friends are fed into the attention layer. The following operation expresses the user's social specific attention on his or her friend in the context of an interactive item:

$$\begin{aligned} H^{u_i} &= ReLU(W_u\widetilde{u_i} + W_cC_i^T + W_v v_j) \\ a^{u_i} &= \frac{exp(H^{u_i})}{\sum_{j\epsilon F(i)} exp(H^{u_j})} \end{aligned}$$

where W_f, W_u, and W_v are the weight matrices of the attention network. Through the fully connected layer, the representation of the user, friends, and item are mapped into the same latent space. ReLU is used as the activation function to increase the nonlinear learning ability of the model. Then, a softmax function, which is often used in an attention neural network, is applied to map the representation H^{u_i}, which is learned from the last layer, onto a number to obtain the attention vector a^{u_i}. a^{u_i} can be viewed as the user's specific attention feature on a specific aspect of the user's friend. Then, the user's friends' features and the user's correlation matrix are fed into the attention layer to calculate the adaptive influence of the friend in the context of the interactive item. Similar to a^{u_i}. The friend's influence attention vector a^{f_i} is obtained by the same process:

$$\begin{aligned} H^{f_i} &= ReLU(W_f F_i + W_u C_i^T + W_v v_j) \\ a^{f_i} &= \frac{W_{h^{f_i}}^T exp(H^{f_i})}{\sum_{j\epsilon F(i)} W_{h^{f_i}}^T exp(H^{f_j})} \end{aligned}$$

By multiplying the obtained attention by the original feature, we can obtain the user's social specific similarity and social influence. Finally, the user's social-based preference feature is obtained by concatenating the user's social specific similarity and social influence and placing it into the fully connected layer. The operation is given as follows:

$$\begin{aligned} user\_specific_i &= \Sigma a^{u_i}\widetilde{u_i} \\ user\_infulence_i &= \Sigma a^{f_i}\widetilde{f_i} \\ u_i = ReLU(W_{us} * user\_spe_i &+ W_{ui} * user\_infu_i + b_s) \end{aligned}$$

4) FUSION LAYER

The fusion layer aims to fuse the feature representations obtained from embedding and co-attentive layers into a synthetic layer. Generally, various fusion strategies can be utilized to fuse different features such as concatenation, addition, and element-wise product. Herein, we first utilize the addition method to fuse the user's item-based representation and social-based representation to obtain the user's complete representation. As the element-wise product method has been proved to be a superior method in the user-item interaction model, the element-wise product method is applied to learn the interaction between them because the element-wise product method may lose some features that may be useful for future user-item interaction learning. The results of the element-wise product are further concatenated with the user and item's representation to obtain the final representation.

5) PREDICTION LAYER

In this task, we need to predict the probability of user u clicking the item i. The representation obtained from the fusion layer will be fed into a fully connected layer to predict the user's rating score as follows:

$$\begin{aligned} h_l &= ReLU(W_l h_{l-1} + b_l) \\ h_{l-1} &= ReLU\ (W_{l-1} h_{l-2} + b_{l-1}) \\ &\cdots \\ h_1 &= ReLU(W_1 h_0 + b_1) \end{aligned}$$

As increasing the number of hidden layers in the neural network might increase the training time and the tendency of "overfitting". In fact, we only used the three-layer network structure, which only contains one hidden layer.

B. LEARNING

1) OPTIMIZATION

Compared to explicit feedback, implicit feedback [4] (such as clicks and buying products) has a broader application prospect. Therefore, our experiment focuses on implicit feedback between the user and item. For implicit feedback, the point-wise loss function and pair-wise loss function are widely used to optimize the personalized ranking recommendation model. Here we concentrate on the pairwise loss function. BPR is a pairwise loss method that has commonly been used in previous studies. It assumes that observed interactions should be ranked higher than unobserved ones. Generally, the objective function (to be minimized) of BPR is defined as:

$$L_{BPR} = \sum_{(u,i,j)\epsilon D_s} -ln\sigma(\check{y}_{ui}(\Theta) - \check{y}_{uj}(\Theta)) + \lambda_\Theta||\Theta||^2$$

where $D_s := (u, i, j | i \epsilon I_u^+ \wedge j \epsilon I \setminus I_u^+)$ denotes a set of pairwise training instances, where item i denotes the items that user u has previously interacted with, and j denotes the items that user u has not interacted with. In practice, for each positive user-item pair, a negative item is randomly sampled from the unobserved item set. In addition, σ is a sigmoid function, and λ_Θ denotes regularization parameters to prevent overfitting.

2) TRAINING

Algorithm 1 shows the training algorithm to optimize the proposed model. First, the parameters are initialized with a Gaussian distribution. As the datasets only contain positive instances, unobserved rating items are randomly sampled as negative instances. Previous works have shown that the optimal sampling ratio for the top-K recommendation is around 4 to 6. Here, the negative sampling ratio is fixed to 4. Using the positive and negative instances, the forwards network is trained to optimize the objective function (presented in Equation2). Then the parameters are updated during the backwards propagation.

Algorithm 1 Optimization Algorithm

Input: R: Observed Rating Matrix; G_u:Users' Social Network;
Output: unobserved rating r^*
Initialize *parameters* : D, λ, dp, lr, num_neg;
for $i = 1$ to *epoch* **do**
Sample mini-batch size user-item pairs;
For each positive pair (u, p_i) and negative pair (u, n_i);
pos_prediction = model(Θ_u,Θ_u);
neg_prediction = model(Θ_u,Θ_u);
Loss(pos_prediction,neg_prediction)
Update Θ_u and Θ_u via backward()
end for

IV. EXPERIMENTS

To analyze and evaluate our proposed method, we conducted a large amount of experiments on four real world data sets to answer the following research questions:

RQ1 Can the proposed method ANSR perform better than the state-of-the-art methods?

RQ2 Do the key components of our proposed method — the co-attentive module and network embedding module — help improve the performance?

RQ3 Can our proposed method alleviate the data sparsity and cold start problems in recommendation tasks?

A. EXPERIMENTS SETTINGS

1) DATASETS

To evaluate the efficacy of ANSR, we conducted experiments on the following four real-world datasets: Delicious, Ciao, Epinions, and Douban. The statistical information of the datasets is summarized in Table1.

TABLE 1. Data set statistics.

Data set	Delicious	Epinions	Ciao	Douban
users	1,521	18,163	17589	2,848
items	1,202	37,325	16,121	39,586
ratings	8,397	374,658	62,452	894,887
social	10,401	287,260	40,133	35,770

- *Delicious:* Delicious is a real-world dataset that contains the users' book ratings and social connections from the Delicious social bookmarking system(http://www.delicious.com). It is published by HetRrc [29].
- *Ciao:* Ciao is a real-world dataset published by Guo *et al.* [23]. It is crawled from a UK's DVD website. It contains the users' movie rating records and trust relations. Since the rating records are explicit feedback and our study focuses on implicit feedback, the rating records are set to 1 as implicit feedback.
- *Epinions:* Epinions is a widely used dataset in recommendation system research. It contains the records of ratings given by users and the trust statements issued by users. It was published on LIBREC's website.
- *Douban:* Douban (https://www.douban.com) is a Chinese social platform where users rate the music, movies, and books that they are interested in. The mechanism of Douban is just like Twitter. The user's rating records and connections are crawled from the movie category to form the experimental dataset. The dataset is published on the website. (https://pan.baidu.com/s/1hrJP6rq#list/path=%2F)

2) BASELINES

To evaluate the performance of our proposed method, we compared our method with the following traditional recommendation methods, social-aware recommendation methods and deep learning-based methods:

- *BPR:* This is a method to optimize the MF method with a pairwise ranking loss. It is a highly competitive baseline for rank recommendation tasks.
- *SBPR:* This is a social-based recommendation method that extends BPR to estimate the user's rankings of items. This model is based on the assumption that users tend to give a higher score to an item that their friends rated.
- *NCF:* This is a deep learning-based method that replaces the inner product of the user and item with a multi-layer perceptron neural architecture to learn the nonlinear interaction between the user and item. It has been confirmed that NCF can achieve a superior performance over traditional recommendation methods.
- *DMF:* This is a new neural matrix factorization model for item ranking recommendation tasks [20]. This model utilizes both explicit feedback and implicit feedback with a neural network architecture to learn the representation of the user and item.

- *CUNE-BPR:* This is a novel collaborative user network embedding method [21] that can extract reliable friends data from user-item feedback and then utilize these reliable friends data to improve social-based recommendations.
- *APR:* This is a novel adversarial personalized ranking (APR) [22] method that built on BPR and incorporates an additional objective function to improve the performance. It achieves the state-of-the-art performance for item ranking recommendations.

3) EVALUATION PROTOCOLS

To evaluate the performance of ANSR, we adopted the leave-one-out protocol, which is a common method in top-K recommendation tasks. For each user, we randomly picked one rating record as the test data, and the rest of the data were used as the training data. As it takes time to rank the test item among all the item data, we randomly sampled 300 items that were not in the rating records for each user and ranked the test item among these items.

As we need to evaluate top-K recommendation tasks, the hit ratio(HR) and normalized discounted cumulative gain(NDCG) are utilized to evaluate the model's performance. The hit ratio (HR@K) measures the percentage of the test items that appears in the top-k list. NDCG (NDCG@K) measures whether the test item has a higher rank in the top-K item list. Therefore, the higher the value of HR and NDCG, the better the model's performance. HR@K and NDCG@K are defined as follows:

$$HR@K = (NumberofHits@K)/(|GT|)$$
$$NDCG@K = Z_k \Sigma_{i=1}^{K}(\frac{2^{r_i}-1}{log_2(i+1)})$$

where, the denominator |GT| is the number of test items, and the numerator is the sum of the test items present in the each user's top-K list. r_i represents the "hierarchical correlation" at position i, which can be treated as 0/1. If the item at position i is in the test set, $r_i = 1$; otherwise, it equals 0. In addition, r_k is the normalized coefficient, which represents the inverse of the sum in the best case of the latter summation formula, to make the value calculated by NDCG within 0-1.

4) PARAMETER SETTINGS

ANSR was implemented based on Pytorch, a popular Python library for deep learning. We randomly selected 20% of the dataset as the validation set to tune the hyper-parameters. The test set was selected by the leave-one-out protocol, which is often used in the top-K ranking task. The initial model parameters were sampled from a zero mean and unit standard deviation Gaussian distribution. The learning rate was tuned among [0.001, 0.005, 0.01, 0.05], and the mini-batch size was tuned among [64,128,256,512]. We adopted the Adam optimizer to optimize the objective function. We adopted ReLU as the activation function between hidden layers to realize a model with a nonlinear learning ability. To avoid overfitting, we tuned the dropout ratio to be between 0.4 and 0.6. The regularization parameter was tuned among [0.01, 0.001, 0.0001, 0.00001].

B. OVERALL PERFORMANCE COMPARISON

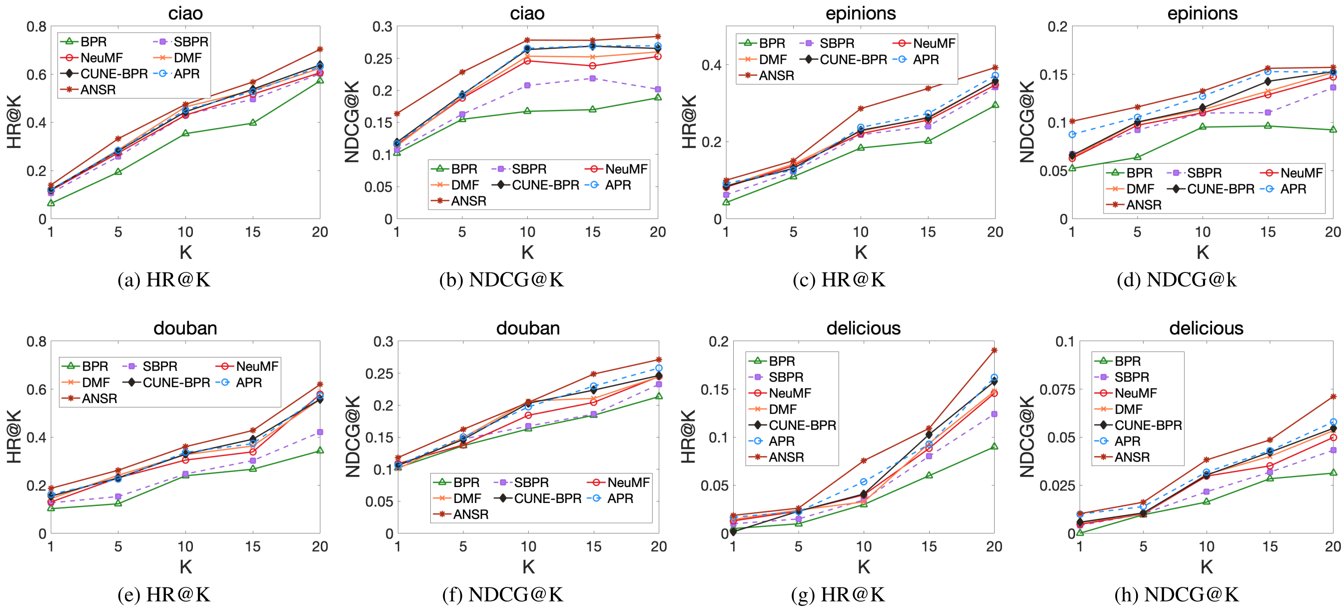
In this section, we compare the performance of our proposed algorithm with those of other baseline algorithms and experiment on four real data sets. Fig.4 compares the HR and NDCG evaluation results for different top-K values in the four datasets. From the results, we can draw the following conclusions. 1) Our proposed method ANSR outperforms other methods. This result confirms the validity of our proposed method. 2) The model based on the deep learning method significantly improves the effect compared to the traditional factor decomposition algorithm, demonstrating the superiority of the deep learning-based recommendation algorithm. This is because the deep learning-based method can learn nonlinear interactions between the user and item. 3) Most models that leverage social connections can improve the performance of the recommendation system compared with models that do not leverage social connections. This is because social-based recommendations can utilize auxiliary information to enrich the user's information, thus solving the problem of data sparsity. Therefore, it can also be proved that utilizing social connections can improve the performance of a recommendation model. 4) Compared with other social-based recommendation methods, our proposed method can effectively extract user information through different social aspects (as mentioned in the method section), so as to learn a better user representation. Therefore, our proposed model performs better than other social-based models.

C. MODEL ANALYSIS

1) CO-ATTENTION INTERACTIVE MODULE ANALYSIS

The overall comparison experiment results demonstrate the effectiveness of our proposed method. To further understand the importance of the co-attention neural network module in the proposed model, we conducted an ablation study. Since the main motivation of our work is to learn how friends influence the user's preference, we compared the complete model with variant models. For one variant model, the attention module was deleted, and the variant model only utilized a network embedding to exploit social connections. For another variant model, the attention module was deleted and the model utilized the uniform weight assignment strategy. For convenience, we used ANSR to represent the complete method, SNE to represent the model only utilizing network embedding, and SNU to represent the model utilizing the uniform weight assignment strategy.

Table.2 shows the performance comparison results on different datasets. From the table, we can see that SNE performs worst. This is because SNE only utilizes network embedding to learn social information to enrich the user's features. Just utilizing a network embedding cannot capture enough social information to learn the user's preference. In addition, directly utilizing social information cannot accurately capture



(a) HR@K (b) NDCG@K (c) HR@K (d) NDCG@k

(e) HR@K (f) NDCG@K (g) HR@K (h) NDCG@K

FIGURE 4. Overall HR@K and NDCG@K performance comparison w.r.t different embedding sizes on the four datasets.

TABLE 2. Comparison of the variant models in the ablation study.

	ciao		epinions	
	HR@10	NDCG@10	HR@10	NDCG@10
SNE	0.4302	0.2597	0.2362	0.1173
SNU	0.4495	0.2602	0.2415	0.1247
ANSR	**0.4751**	**0.2614**	**0.2860**	**0.1326**
	delicious		douban	
	HR@10	NDCG@10	HR@10	NDCG@10
SNE	0.0573	0.0265	0.3251	0.1863
SNU	0.0677	0.0320	0.3435	0.1926
ANSR	**0.0756**	**0.0382**	**0.3607**	**0.2044**

TABLE 3. Comparison of different network embedding modules in the proposed model.

	ciao		epinions	
	HR@10	NDCG@10	HR@10	NDCG@10
_GD	0.4326	0.2598	0.2402	0.1198
_hope	0.4598	0.2603	0.2751	0.1287
_n2v	0.4602	0.2607	0.2785	0.1292
_line	0.4632	0.2602	0.2815	0.1307
_snde	**0.4751**	**0.2614**	**0.2860**	**0.1326**
	delicious		douban	
	HR@10	NDCG@10	HR@10	NDCG@10
_GD	0.0591	0.0229	0.3202	0.1673
_hope	0.0708	0.0340	0.3535	0.1947
_n2v	0.0713	0.0347	0.3541	0.1972
_line	0.0723	0.0362	0.3581	0.2017
_snde	**0.0756**	**0.0382**	**0.3607**	**0.2044**

the changes in the user's preferences due to social relations. SNU performs better than SNE. Obviously, AN-uniform can capture the user's preference influenced by social relations to some extent. However, it is based on the assumption that the user's social network is homogeneous. In the real world, the user's preferences are not always the same as their social friends. In particular, different friends have different influences on users' preferences when the user is interacting with different friends. Therefore, ANSR performs best because it can capture two special social features from the viewpoint of social effects. One social special feature can learn the user's special attention on one aspect of the user's friend. Another social special feature could learn the adaptive influence of friends on the user to help improve the model's accuracy.

2) NETWORK EMBEDDING MODULE ANALYSIS

In the previous work, the user's embedding was initialized from a Gaussian distribution with a zero mean and standard deviation. As we mentioned in Section 3, the network embedding can learn a better node embedding from a low-dimensional latent space while maintaining the network topology structure and network content information. Therefore, we conducted an ablation study to learn the effect of the network embedding module in our proposed model. Although there are different network embedding methods to learn the node embedding from a social network, herein, we compared the performance of utilizing different network embedding methods in our model. For convenience, we used _GD to represent the variant model that only utilizes the Gaussian distribution initialization strategy, and _[method] to represent the model that utilizes different network embedding strategies. For example, _snde represents the model that utilizes the SNDE network embedding strategy. Table.3 compares the performance of the variant models. We can observe that utilizing network embedding as a pre-trained strategy obviously outperforms _GD, which only utilizes a Gaussian distribution initialization strategy. Among them, _snde provides the best result. This may be because SNDE can maintain the similarity between two connected nodes in the network, while conserving the secondary similarity between two nodes with common neighbors that are not directly connected. From a real-world complex network perspective, although there is no direct connection between two users, they may be in the same community structure and their preferences may be similar. Therefore, the node embedding learned through SNDE can maintain more similar information and perform better. In addition, we also found that utilizing network embedding as a pre-trained strategy can accelerate the convergence speed in the experiment.

D. DATA SPARSITY PROBLEM STUDY

As our main purpose is to utilize social connections to alleviate the data sparsity problem, we performed comparative experiments to analyze whether our proposed model can effectively alleviate this issue. To investigate this problem, we change our data sets to the data sets with different data sparsity levels. First, we divided the users into four different groups according to the data sparsity. For example, Douban_5 is the dataset where the user has 1-5 interaction rating records, Douban_10 has 5-10 interaction rating records, etc. Then, we compared the performance of our proposed model to the deep learning-based and social-based recommendation baseline methods like NeuMF and CUNE-BPR to investigate how our proposed model performs using different sparsity datasets.

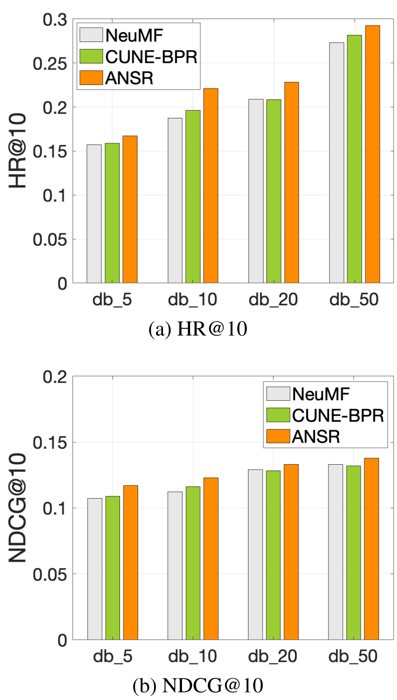


FIGURE 5. (a), (b) are HR@10 and NDCG@10 of NeuMF, CUNE-BPR, ANSR on the Douban dataset w.r.t. different data sparsity levels.

Fig.5 shows the performance comparison results. As the data sparsity changed, our proposed model always performed better. In particular, compared with the algorithm utilizing only rating feedback, our algorithm has a greatly improved performance because it can effectively use more social information to solve the problem of data sparsity. However, with the increasing of data sparsity, the performance improvement becomes smaller, which may be because when the rating data is rich enough, social relations can only serve as auxiliary information to help improve the prediction accuracy of the model.

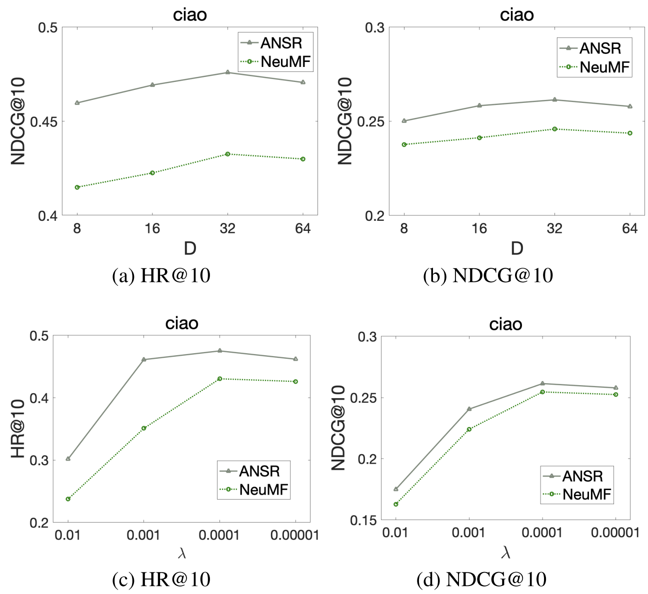


FIGURE 6. (a) and (b) are HR@10 and NDCG@10 of ANSR on the Ciao dataset w.r.t different embedding dimension size D, respectively. (c) and (d) are HR@10 and NDCG@10 of ANSR on the Ciao dataset w.r.t different regularization parameter λ, respectively.

E. PARAMETER SENSITIVITY STUDY

In this section, we will analysis the influence of some hyper-parameters on the performance of the proposed model. The effect of the dropout ratio is similar to the regularization parameter, which is utilized to constrain the model to prevent overfitting. Therefore, we mainly discuss the impact of embedding dimension size D and regularization parameters λ on model performance. Fig.6 shows the performance comparison under different parameters. From the figure, we can draw the following conclusions. 1) With the increasing of the embedded dimension size D, the performance of the model gradually improves, and when the embedded dimension increases to 64, the performance starts to decrease. This is because when the embedding dimension is too small, the model lacks generalization ability. However, if the embedding dimension is too large, the weight parameters of the model will be too large, which will increase the complexity of the model and lead to its performance degradation. Therefore, selecting an appropriate embedding dimension size can improve the learning ability of the model. 2) Selecting appropriate regularization parameter λ plays a decisive role in the model. Because only a small number of features in the sample are critical to the final prediction model, when the regularization parameter λ is too small, it fails to constrain the nonessential features, and when the weight parameter λ is too large, the key weight feature for prediction will be suppressed. Therefore, choosing a reasonable regularization parameter λ can not only improve the prediction ability of the model, but also improve the generalization ability of the model.

V. CONCLUSION

In this paper, we first explore the reasons that affect the performance of the recommendation system. Based on these reasons, we introduced two special social features that can model how is the user's preference affected by the user's social connections and designed a deep recommendation model based on an attention neural network to predict the user's unknown rating of the item. It is shown that the proposed model can extract users' social features that can improve the prediction accuracy. Extensive experiments were conducted on four datasets to verify the validity of our proposed model.

For future work, we will improve the model's interpretability. The recommendation system can not only recommend the items that the user is interested in but also explain the reasons for the recommendation to improve the user's trust in the recommendation system.

REFERENCES

[1] M. Li, K. Tei, and Y. Fukazawa, "An efficient co-attention neural network for social recommendation," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Oct. 2019, pp. 34–42.

[2] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 135–142.

[3] A. J. B. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *Proc. 9th ACM Conf. Recommender Syst. (RecSys)*, 2015, pp. 43–50.

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

[5] S. Fazeli, B. Loni, A. Bellogin, H. Drachsler, and P. Sloep, "Implicit vs. explicit trust in social matrix factorization," in *Proc. 8th ACM Conf. Recommender Syst. (RecSys)*, 2014, pp. 317–320.

[6] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 325–334.

[7] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, 2010, pp. 241–250.

[8] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*, 2016, pp. 233–240.

[9] Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, "Joint representation learning for top-N recommendation with heterogeneous information sources," in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2017, pp. 1449–1458.

[10] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*. [Online]. Available: http://arxiv.org/abs/1511.06939

[11] He, Xiangnan, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web Int. World Wide Web Conf. Steering Committee*, 2017, pp. 173–182.

[12] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanhalli, "A^{3}NCF: An adaptive aspect attention model for rating prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3748–3754.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 701–710.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 855–864.

[15] 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 (WWW)*, 2015, pp. 1067–1077.

[16] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, pp. 1225–1234.

[17] E. Zhong, W. Fan, and Q. Yang, "User behavior learning and transfer in composite social networks," *ACM Trans. Knowl. Discovery from Data*, vol. 8, no. 1, pp. 1–32, Feb. 2014.

[18] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "ETAF: An extended trust antecedents framework for trust prediction," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 540–547.

[19] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2014, pp. 261–270.

[20] H.-J. Xue, X. Dai, J. Zhang, S. Huang, and J. Chen, "Deep matrix factorization models for recommender systems," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3203–3209.

[21] C. Zhang and L. Yu, "Collaborative user network embedding for social recommender systems," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2017, pp. 353–362.

[22] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2018, pp. 355–364.

[23] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2016, pp. 5–14.

[24] C. Zhang, L. Yu, and Y. Wang, "Collaborative user network embedding for social recommender systems," in *Proc. SIAM Int. Conf. Data Mining Soc. Ind. Appl. Math.*, 2017.

[25] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[26] S. Purushotham, Y. Liu, and C.-C. Jay Kuo, "Collaborative topic regression with social matrix factorization for recommendation systems," 2012, *arXiv:1206.4684*. [Online]. Available: http://arxiv.org/abs/1206.4684

[27] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017.

[28] 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 (WSDM)*, 2012, pp. 173–182.

[29] I. Cantador, P. L. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)," in *Proc. 5th ACM Conf. Recommender Syst.*, 2011, pp. 387–388.

MUNAN LI received the B.S. degree in software engineering from the Dalian University of Technology, in 2013, and the M.S. degree in computer science from the China University of Petroleum, in 2016. She is currently pursuing the Ph.D. degree in computer science with Waseda University, Tokyo, Japan.

Her research interests include machine learning, deep learning, recommender systems, and complex networks.

KENJI TEI received the B.E and D.E degrees in information and computer science and the Ph.D. degree in engineering from Waseda University, Japan, in 2003, 2005, and 2008, respectively.

He joined the Department of Information and Computer Science, Waseda University, as a Research Assistant. From 2008 to 2010, he was an Assistant Professor with the Media Network Center, Waseda University, and a Project Assistant Professor with the National Institute of Informatics. He was an Assistant Professor and an Associate Professor with the National Institute of Informatics and SOKENDAI, from 2010 to 2015 and from 2015 to 2018, respectively. Since 2018, he has been an Associate Professor with Waseda University, where he is currently an Associate Professor with the Faculty of Science and Engineering. He research interest includes software engineering for self-adaptive software, in particular, models@runtime techniques and software architecture for self-adaptive software.

YOSHIAKI FUKAZAWA (Member, IEEE) received the B.E, M.E., and D.E. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1976, 1978, and 1986, respectively.

He joined the School of Science and Engineering, Waseda University, as a Research Associate, in 1978, the Department of Computer Science, Sagami Institute of Technology, as a Lecturer, in 1983, and the Department of Electrical Engineering, School of Science and Engineering, Waseda University, as an Associate Professor, in 1987. He is currently a Professor with the Department of Information and Computer Science, Waseda University. He is also the Director of the Institute of Open Source Software, Waseda University. His research interest includes software engineering, especially reuse of object-oriented software, agent-based software, and software optimization.

Dr. Fukazawa is a member of IPSJ, JSSST, IEICE, and ACM.