A Joint Neural Model for User Behavior Prediction on Social Networking Platforms

JUNWEI LI, LE WU, RICHANG HONG, and KUN ZHANG, Hefei University of Technology
YONG GE, The University of Arizona
YAN LI, Beijing Kuaishou Technology Company Limited

Social networking services provide platforms for users to perform two kinds of behaviors: consumption behavior (e.g., recommending items of interest) and social link behavior (e.g., recommending potential social links). Accurately modeling and predicting users' two kinds of behaviors are two core tasks in these platforms with various applications. Recently, with the advance of neural networks, many neural-based models have been designed to predict a single users' behavior, i.e., social link behavior or consumption behavior. Compared to the classical shallow models, these neural-based models show better performance to drive a user's behavior by modeling the complex patterns. However, there are few works exploiting whether it is possible to design a neural-based model to jointly predict users' two kinds of behaviors to further enhance the prediction performance. In fact, social scientists have already shown that users' two kinds of behaviors are not isolated; people trend to the consumption recommendation of friends on social platforms and would like to make new friends with like-minded users. While some previous works jointly model users' two kinds of behaviors with shallow models, we argue that the correlation between users' two kinds of behaviors are complex, which could not be well-designed with shallow linear models. To this end, in this article, we propose a neural joint behavior prediction model named *Neural Joint Behavior Prediction Model (NJBP)* to mutually enhance the prediction performance of these two tasks on social networking platforms. Specifically, there are two key characteristics of our proposed model: First, to model the correlation of users' two kinds of behaviors, we design a fusion layer in the neural network to model the positive correlation of users' two kinds of behaviors. Second, as the observed links in the social network are often very sparse, we design a new link-based loss function that could preserve the social network topology. After that, we design a joint optimization function to allow the two behaviors modeling tasks to be trained to mutually enhance each other. Finally, extensive experimental results on two real-world datasets show that our proposed method is on average 7.14% better than the best baseline on social link behavior while 6.21% on consumption behavior prediction. Compared with the pair-wise loss function on two datasets, our proposed link-based loss function improves at least 4.69% on the social link behavior prediction and 4.72% on the consumption behavior prediction.

This research is partially done during the first author's internship in Beijing Kuaishou Technology Company Limited.
This work is supported in part by grants from the National Natural Science Foundation of China (Grant No. 61972125, U1936219, 61722204, 61932009), the Fundamental Research Funds for the Central Universities (Grant No. JZ2020HGPA0114), and Zhejiang Lab (No.2019KE0AB04).
Authors' addresses: J. Li, L. Wu (corresponding author), R. Hong, and K. Zhang, Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei, Anhui 230601, China; emails: {lijunwei.edu, lewu.ustc, hongrc.hfut, zhang1028kun}@gmail.com; Y. Ge, Eller College of Management, University of Arizona, Tucson, AZ 85721; email: yongge@email.arizona.edu; Y. Li, Department of Multimedia Understanding, Beijing Kuaishou Technology Company Limited, Beijing 100085, China; email: liyan@kuaishou.com.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2157-6904/2020/09-ART72 $15.00
https://doi.org/10.1145/3406540

CCS Concepts: • **Information systems** → *Retrieval models and ranking*; *Recommender systems*; *Data mining*; *Personalization*;

Additional Key Words and Phrases: Joint neural networks, social link behavior, consumption behavior, behavior prediction, topology information

ACM Reference format:
Junwei Li, Le Wu, Richang Hong, Kun Zhang, Yong Ge, and Yan Li. 2020. A Joint Neural Model for User Behavior Prediction on Social Networking Platforms. *ACM Trans. Intell. Syst. Technol.* 11, 6, Article 72 (September 2020), 25 pages.
https://doi.org/10.1145/3406540

1 INTRODUCTION

Social Networking Services (SNSs) provide platforms for users to connect with others and share their preferences with social friends. For example, in the popular online platform *Epinions*, a user follows people she/he trusts. Meanwhile, the user's preferences for items would be displayed to social followers. With the arrival of the era of big data, how to utilize advanced data analysis and information science to better understand people's social lives online and handle the social and juristic challenges on technology become very important [17, 38]. For SNSs platforms, there are many users and items interacting with each other. Thus, accurately modeling users' two kinds of behaviors: social link behavior (i.e., recommending possible social links) and consumption behavior (i.e., recommending potential items of interest) with insensitive information have become two core tasks, which could benefit many applications in our real world, such as item recommendation [37, 43] and social link suggestion [30].

In the past, these two kinds of users' behaviors have been extensively explored. On the one hand, social link behavior prediction models utilized the user-user social network topology to predict the potential proximity between two users [30]. On the other hand, latent factor based Collaborative Filtering (CF) models are popular for users' item preference prediction [1, 37, 41]. With user-item consumption behavior, these latent factor based models associate each user and each item with a latent vector in a latent space. Then, the predicted preference score of a user to an item is approximated by the inner product operation of corresponding latent vectors. Similarly, by treating the user-user link recommendation as a matrix factorization task, researchers also adopted collaborative filtering on social link behavior prediction [35]. However, as each user only connected to limited social users and interacted with several items, both of these two behavior prediction tasks suffer from the data sparsity problem [24], making a suboptimal performance. In the SNS platform, social scientists have long converged that users' two kinds of behaviors are not isolated. On the contrary, they are highly correlated with each other. People like to connect to others that show similar item consumption behavior. Also, they are easily influenced by their social neighbors and perform similar consumption behavior. Based on this phenomenon, many works have been proposed by adopting one kind of behavior to enhance the other [3, 20, 46]. These models could partially alleviate the data sparsity issue and improve model performance. However, most of them only considered the shallow linear interactions of user-item and user-user pairs by using the simple and fixed inner product, which limited the performance of models [16]. Therefore, the complex interactions among social entities need to be further studied.

Recently, deep neural networks have shown great potential for automatic representation learning and deliver state-of-the-art performance on various domains [23, 36, 45]. Researchers have been in a race applying deep learning based techniques for user behavior modeling. For CF, researchers argued that the user-item interaction is rather complex, and could not be well captured by the shallow linear interaction function between a user and an item with classical latent factor based models. Therefore, many neural network models were proposed to fit more complex interactions between users and items [9, 13, 16]. The key idea is to adopt deep neural networks to extract deep non-linear features. Generally, these deep neural network models perform better than shallow latent factor CF models. For example, Neural Matrix Factorization (NeuralMF) has shown state-of-the-art performance by modeling each user's consumption behavior with two parts: a shallow linear interaction part as traditional CF models, and a deep neural interaction part that can fit the non-linear, complex relationship between users and items [16]. Nevertheless, they only focused on these two behaviors separately and ignored the correlations between different behaviors.

In fact, all the abovementioned models (both the shallow models and the neural-based models) aim at predicting a particular kind of users' behavior. However, users' two kinds of behaviors are closely related. As stated before, users' consumption behavior may be affected by their social relationships, leading to similar consumption interests among socially connected users [20, 22]. Besides, users would like to connect with others who share similar taste [46, 59]. Therefore, by jointly modeling the users' two kinds of behaviors, these two tasks could mutually enhance each other and further improve the performance of both prediction tasks. Specifically, Friendship and Interest Propagation (FIP) was proposed to jointly model the users' two kinds of behaviors by sharing user latent representation over users' two kinds of behaviors [59]. Besides, as users' behaviors evolve over time, Wu et al. proposed the shallow model Evolving Joint Prediction (EJP) to model the social influence and the homophily effect over time [52]. Furthermore, Neural Joint Modeling (NJM) modeled the evolution of items over time by recurrent neural networks [57]. However, this model heavily relies on the input data and could not be easily adapted when the temporal data is not available. Therefore, it is natural to ask whether it is possible to design a general model for two kinds of behaviors.

Concretely, there remain two key challenges for the above questions. First, different from single behavior prediction, there are interactions between social link behavior and consumption behavior. Thus, it is challenging to model the mutual correlations between them in a joint unified framework. Second, users in a social network have topological relationships, which means that two users who are not connected in a social network may become friends because of their similar topological information (such as having similar social neighbors). Besides, the social network differs from other kinds of auxiliary data (images or texts) that can be pre-trained. It will be influenced by users' consumption interests and the inner linked relations itself. Therefore, how to inject all of the social impact property in a unified neural framework to enhance recommendation is also a challenge.

To tackle the key challenges mentioned above, in this article, we explore users' two kinds of behaviors on SNSs and propose a neural joint behavior prediction model. Specifically, there are two key characteristics of our proposed model: First, to model the correlation of users' two kinds of behaviors, we design the fusion layer in the neural network to model the correlation of users' two kinds of behaviors. Second, in the social link behavior modeling process, we design a new link-based loss function that could not only model users' historical social link behavior but also could preserve the network topology. After that, we design a joint optimization function to make these two behavior modeling tasks mutually enhance each other. Finally, extensive experimental results on two real-world datasets clearly show the effectiveness of our proposed model.

2 RELATED WORK

In this section, we will introduce the related work, which can be grouped into three categories: (1) social link behavior prediction methods, (2) the models for consumption behavior prediction, and (3) the works for modeling users' two kinds of behaviors.

Social Link Behavior Prediction. Social link behavior prediction is a special case of link prediction that aims at predicting potential new connections among a social network in a near future [30] and has been studied by many researchers in our real life scenes, such as Twitter [33]. Traditional unsupervised methods are based on the pre-defined similarity or topological property between two nodes, for example, the widely used common neighbors [31], Leicht-Holme-Newman index [25], and node rank based algorithms [21]. In contrast to the unsupervised methods, the supervised models regard the observable links as positive samples while regarding unobserved links as negative samples [35] and they usually perform better than unsupervised methods.

However, in the real world, as each user only connects to limited social users, the user-user network is very sparse. In order to solve this problem, some researchers leveraged side information to alleviate data sparisty. For example, Tang et al. assumed consumption behavior would influence social link behavior. Thus, they proposed hTrust for trust prediciton [46]. Similarly, Beigi et al. suggested that emotions are very important when building relationships among users; they considered emotional impact when performing trust prediction [5]. Besides, some researchers argued that social networks are not static. With the evolution of social networks, researchers also leveraged temporal information in the link prediction task [2].

Different from the above methods, many graph embedding models have emerged in recent years [12, 39]. These works aim at learning low-latent representations of nodes with the principle of preserving the topology information. Most of them not only model the connected first-order links (neighborhood relationship) but also consider the higher-order proximity (topological information of a network). These learned low-latent representations are friendly to subsequent applications, and they have shown better performance in the link prediction task [10].

Nowadays, with the development of deep learning, researchers are now focusing on applying artificial neural networks [26]. In [61], researchers proposed an autoencoder-based approach for link prediction. Besides, due to the complex and heterogeneous characteristics of social networks, Cao et al. [7] aggregated many networked tasks and proposed a unified model called Multi Neural Network (MNN), which assigned an individual neural network for each task to enhance the performance.

Also, to better preserve higher-order information, many network embedding methods also have taken the advantages of deep learning. Wang et al. preserved the topology information by the autoencoder [47], and it can be deemed as a deep neural network based graph embedding method. In order to learn better representations, Liao et al. leveraged the advantages of deep network and proposed Social Network Embedding (SNE) [29], which preserved both the structural proximity and attribute proximity in a social network. To fit various final tasks for networked data, Chang et al. designed Heterogeneous Network Embedding (HNE) [8]. Specifically, HNE utilized a highly nonlinear multilayered embedding function to capture the complex interactions between the heterogeneous data in a network. These deep neural network models indicated a promising approach to link prediction task.

Consumption Behavior Prediction. Users' consumption behavior prediction aims at matching the potential items for users, and it can be deemed as the recommendation task. As one of the most successful methods on recommender systems, Collaborative Filtering (CF) captures the preferences of users by collecting the past historical record from the like-minded users with similar consumption behavior [43]. Among all CF techniques, latent representation models are very popular [27, 32, 37, 53], which embed users and items into a low latent space and describe the preference by the inner product operation.

However, as each user usually interacts with several items, the consumption behavior prediction also suffers from the data sparsity problem. In fact, users are likely to be influenced by their social friends when making consumption decision [34]. Thus, some researchers leveraged social information to boost interest prediction. For example, Jamali et al. proposed SocialMF, which considered the social neighborhood information to enhance the collaborative filtering [20]. Jiang et al. combined social contextual information for recommendation [22]. Wu et al. utilized auto-encoder to extract features from social networks and injected them into a neural network for a better consumption behavior prediction [55]. Because of adopting side social information, these methods usually have better performance than other shallow models that only considered the user-item interactions. However, there are still many limitations when fitting the complex interactions between users and items.

With the gradual popularity of deep learning and due to its great success in other fields (e.g., computer vision [23]), more and more researches have begun to focus on deep learning based recommender systems [50, 56, 62] in recent years. Compared with previous shallow latent representation models, deep learning based methods have some advantages. First, compared to the traditional CF methods that model the user preference with the linear user-item interactions, deep learning based methods are capable of extracting high-level deep feature from users and items. Therefore, He et al. proposed NeuMF to combine the deep neural networks and the matrix factorization model [16]. Based on this work, Bai et al. further proposed Neighborhood-based Neural Collaborative Filtering (NNCF) [4], which aggregated the neighborhood information of users and items. Xue et al. proposed DeepMF [58] to learn the deep features of users and items from one-hot vectors. Overall, these methods use deep neural networks to extract higher-level features. Second, the complex interactions between users and items can be elaborately designed with deep neural networks. For example, He et al. proposed neural factorization machine [15] to incorporate deep neural networks with the Factorization Machine [40] in a unified framework. In industry, Wide & Deep [9] combined both regression and classification task simultaneously. Thus, it could be adapted to a wide range of application scenarios. Guo et al. proposed DeepFM to model low-order feature and high-order feature simultaneously, so that DeepFM could learn the combinatorial relationships among different order features [13]. On the basis of DeepFM, xDeepFM improved the DeepFM by proposing the Compressed Interaction Network (CIN), which compensates for some drawbacks of feature operations on deep neural network [28]. Third, deep learning based methods are flexible to additional information. For example, as a social network and users' preference evolve over time, RRN [49] and ARSE [45] associated temporal information by utilizing the recurrent neural network, they could capture dynamics characteristics in the users and items.

Recently, Graph Convolutional Networks (GCNs) have shown promising ability on consumption behavior prediction task by treating users and items as nodes on a user-item network [6, 60]. As GCNs aggregate information from network neighborhood and stack many layers, it can extract more expressive representations for nodes to preserve the topological information. Based on this idea, Neural Graph Collaborative Filtering (NGCF) extended Graph Convolutional Matrix Completion (GC-MC) [6] with multiple layers, making the higher-order collaborative signals between users and items could be captured in the embedding learning process [48]. Also, because of the strong correlation between two kinds of behaviors, Wu et al. took the advantages of GCNs to extract user embeddings containing transitive information from the social network for social recommendation [54]. Different from these works, our proposed method takes into account the prediction of two behaviors at the same time. And we focus on how to model the correlation between users' different behaviors. Therefore, our model can be seamlessly replaced by GCN-based or other advanced feature extraction approaches.

Modeling Users' Two Kinds of Behaviors. In the real world, sociologists have long converged that users' social link behavior and consumption behavior are not independent and unrelated. Instead, they coexist and interact with each other [3, 34]. For example, a consumption behavior often comes from the recommendation of friends on social platforms, or we would like to make new friends with someone because of sharing the same interests. Based on this phenomenon, many works have combined these two kinds of behaviors to enhance the recommendation. Yang et al. proposed a unified framework Friendship and Interest Propagation (FIP), which utilized the observable properties of users and items and shared users' latent representations to model users' interests and friendships simultaneously [59]. Wu et al. associated the temporal information to explore the evolution of users' preferences and social relations on SNSs with latent-based joint evolution models [51, 52]. These two models predicted users' two behaviors with shallow interactions. Recently, researchers extended these shallow models and proposed a neural joint evolution model NJM [57] for jointly predicting these behaviors with a recurrent neural network structure. The key idea is to model the evolution of users' preferences and items' attributes with a recurrent deep network. This model showed better performance than the shallow joint evolution models. However, NJM could not be applied to the situation when the temporal information is not available. Therefore, there is still much room for improvement on users' multi-behavior prediction. In order to adapt to general data, in this article, we intend to design a novel neural based joint model.

3 PRELIMINARIES AND PROBLEM FORMULATION

3.1 Preliminaries

For ease of presentation, we use lowercase alphabets (e.g., a) to denote scalars, boldface and italic lowercase alphabets (e.g., $\boldsymbol{a}$) to denote vectors, italic uppercase alphabets (e.g., A) to denote set, and boldface uppercase alphabets (e.g., $\mathbf{A}$) to denote matrices. Throughout, we will use i, j, k to denote the users and a, b, c to denote the items.

Matrix Factorization. Matrix factorization (MF) is a widely used shallow model for recommender systems [37]. The principle associates users and items with real-valued latent representations and measures the similarity between users and items by inner product. For example, let $\boldsymbol{u}_i$, $\boldsymbol{u}_j$ and $\boldsymbol{v}_a$ denote the latent representations for user i, user j, and item a, respectively. Then, MF-based models estimated interactions between the user-user pair and the user-item pair as:

$$\hat{t}_{ij} = f(i, j|\boldsymbol{u}_i, \boldsymbol{u}_j) = \boldsymbol{u}_i^T \boldsymbol{u}_j, \tag{1}$$

$$\hat{\mathrm{r}}_{\mathrm{ia}} = f(i, a|\boldsymbol{u}_i, \boldsymbol{v}_a) = \boldsymbol{u}_i^T \boldsymbol{v}_a, \tag{2}$$

where $\hat{\mathrm{t}}_{\mathrm{ij}}$ is the predicted scores of user i to user j, and $\hat{\mathrm{r}}_{\mathrm{ia}}$ is the predicted scores of user i to item a.

As a shallow latent representation model, matrix factorization captures relationships by the simple linear and fixed inner product. Due to the MF's limitation [16], shallow MF-based models are not able to fit the complex interactions. Therefore, researchers resort to deep neural networks to address the limitation. The key idea is to utilize deep neural networks to extract high-level deep feature. However, deep neural networks often lead to overfitting on the test set. Therefore, many methods combined both of them to strike a balance and achieved a better performance [9, 15, 16, 28].

Neural Matrix Factorization Model. As one of the successful deep neural networks based models, NeuMF predicts user-item interactions by combining the deep neural networks and the matrix factorization [16]. The structure of NeuMF is illustrated in Figure 1.

Instead of modeling the user-item preference as the linear interaction function, NeuMF adopted the Generalized Matrix Factorization (GMF) to get the shallow GMF layer as:

$$\phi^{GMF} = \boldsymbol{u}_i \odot \boldsymbol{v}_a, \tag{3}$$

where $\odot$ denote the element-wise product operation.

On the right channel, as neural networks can theoretically approximate most continuous functions [18], NeuMF employed Multilayer Perceptron (MLP) to extract high-level deep interaction

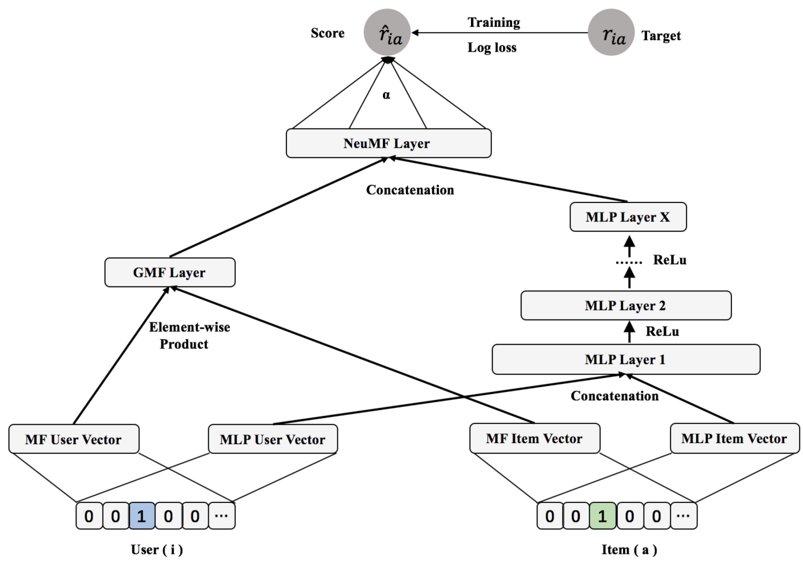


Fig. 1. Neural matrix factorization model [16]. Combining the deep neural networks and the shallow factorization matrix method to make the model have better generalization performance.

features from user-item pairs. More precisely, the MLPs under NeuMF framework is defined as:

$$\begin{aligned} \mathbf{z}_1 &= \phi_1(\boldsymbol{u}_i, \boldsymbol{v}_a) = \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_a \end{bmatrix}, \\ \phi_2(\mathbf{z}_1) &= a_2\left(\mathbf{W}_2^T \mathbf{z}_1 + \mathbf{b}_1\right), \\ &\ldots\ldots \\ \phi^{MLP} &= \phi_L(\mathbf{z}_{L-1}) = a_L\left(\mathbf{W}_L^T \mathbf{z}_{L-1} + \mathbf{b}_L\right), \end{aligned} \tag{4}$$

where $\mathbf{W}_x$, $\mathbf{b}_x$, and a_x denote the weight matrix, bias vector, and activation function for the x^{th} layer's perceptron, respectively.

At last, NeuMF Layer combined the GMF layer and MLP layer to predict the score.

3.2 Problem Definition

On social networking services, given a set of N users $U = \{u_1, u_2, \ldots, u_\mathrm{N}\}$ and a set of M items $V = \{v_1, v_2, \ldots, v_\mathrm{M}\}$, we define a social network matrix as $\mathbf{T} \in \mathbb{R}^{N \times N}$. If user i trusts user j, the $(i, j)^{th}$ entry t_{ij} of $\mathbf{T}$ equals 1; otherwise, it equals 0. Besides, we define a shortest distance table $\mathbf{D}$, in which d_{ij} means the shortest path length from user i to user j. Specifically, it is equal to ∞ if there is no path between them. Similarly, we use $\mathbf{R} \in \mathbb{R}^{N \times M}$ to denote the rating matrix and r_{ia} represents the score that user i rated on item a.

Due to the superior performance of combining deep feature and shallow feature in NeuMF, we also represent each user and each item in two spaces for the consumption behavior: a shallow consumption user latent space $\mathbf{W}$ and shallow consumption item latent space $\mathbf{M}$. Besides, we also project users and items in the deep latent consumption space, where $\mathbf{P}$ denotes deep consumption user latent space and $\mathbf{Q}$ denotes deep consumption item latent space. Similarly, for the social link behavior, we assign deep social user latent space $\mathbf{S}$ and shallow social user latent space $\mathbf{X}$. For ease of notations, we list all the latent spaces in Table 1 for users and items, and we use corresponding boldface and italic lowercase alphabet to denote the deep or shallow feature.

Table 1. Main Notations of Different Latent Spaces

Notations	Interpretation
S	Deep social user latent space
P	Deep consumption user latent space
Q	Deep consumption item latent space
X	Shallow social user latent space
W	Shallow consumption user latent space
M	Shallow consumption item latent space

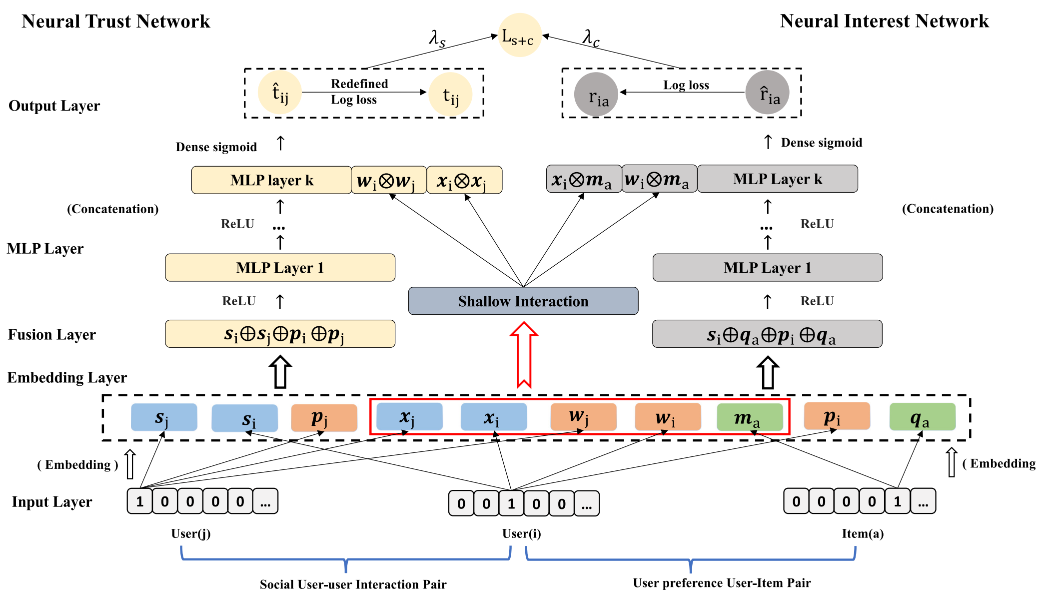


Fig. 2. The structure of the proposed model. Left: the Neural Trust Network predict potential social friends to users. Right: the Neural Interest Network recommends items of interest to users. These two networks utilize the fusion layer to model the correlations between users' two kinds of behaviors and combine both deep neural networks and shallow generalized factorization matrix to make the feature in the output layer more robust.

Based on the terminologies described above, we formally define the problem of our model as:

Problem Definition. Given a social network matrix **T** and the corresponding rating matrix **R**, our goal contains two aspects: (1) Predicting possible social link behavior in a social network, and (2) Predicting users' consumption behavior.

4 THE PROPOSED MODEL

In this section, we first give a brief introduction of our proposed model Neural Joint Behavior Prediction Model (NJBP), which contains two key components: the *Neural Trust Network* and *Neural Interest Network*. Then, we introduce the structure and technical details of these two key components of NJBP. The overall architecture of NJBP is presented in Figure 2,

Generally speaking, NJBP is composed of two parts: the *Neural Trust Network* is on the left channel for social link behavior prediction, and the *Neural Interest Network* is on the right channel for consumption behavior prediction. At the bottom of NJBP, there is an input layer, which reads the

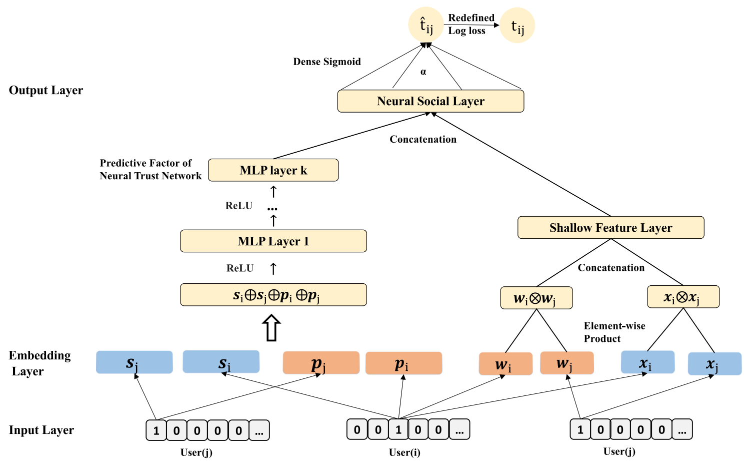


Fig. 3. The structure of the Neural Trust Network. Left: the deep neural networks extract deep interaction features. Right: the shallow generalized factorization matrix makes the model more robust. The fusion features existing in both deep and shallow parts have modeled the correlation and interaction between different behaviors.

one-hot encoding ID of users and items. After mapping the one-hot vectors in input layer, we get the embedding vectors of users and items. The embedding vectors represent the characteristics of users and items and will be learned later. Then, we purposefully fuse deep and shallow embedding vectors in the fusion layer. Next, the fused features get to the MLP layer, which performs non-linear projection to extract the deep features from the input of MLP. Finally, we design a link-based loss function in the *Neural Trust Network* to preserve the topological information of a social network, and in the output layer, we combine high-level deep features and shallow MF features to predict the behaviors of users. We detail each part in the following sections.

4.1 Neural Trust Network

The overall architecture of *Neural Trust Network* is shown in Figure 3. It aims at modeling users' social link behavior. In this part, we give detailed explanations about how to model the social relationships with the correlation of users' interest.

Input Layer. We follow the operation of other behavior prediction tasks [7, 16]; the input is the sparse social network matrix **T**. To keep track of the total N users, a simple yet commonly adopted idea is to represent each user with a one-hot encoding vector. These one-hot encoding vectors will be regarded as identities of users to select corresponding representations in the next layer.

Embedding Layer. Embedding layer contains the low latent representations of users. Specifically, the lookup operation will be used to select the corresponding row from the deep social user latent space S and the deep consumption user latent space **P**. As shown in Figure 3, for user i and user j, we get raw deep representations $\boldsymbol{s}_i$, $\boldsymbol{p}_i$, $\boldsymbol{s}_j$, and $\boldsymbol{p}_j$. To capture the shallow MF representations, we also select the corresponding row from the shallow social user latent space **X** and the shallow consumption user latent space **W**; then, we get the shallow representations $\boldsymbol{x}_i$, $\boldsymbol{w}_i$, $\boldsymbol{x}_j$ and $\boldsymbol{w}_j$.

Fusion Layer. We design the fusion layer to capture the shallow linear user-user interactions and the correlations between different behaviors. Specifically, we fuse users' representations from two aspects: (1) For deep latent spaces, we have: $\boldsymbol{s}_i \oplus \boldsymbol{s}_j$ and $\boldsymbol{p}_i \oplus \boldsymbol{p}_j$, where $\oplus$ is the concatenate operation on two vectors, since social relationships may be influenced by social network itself and the shared interests. (2) For shallow latent spaces, inspired by other deep neural models [4, 16], we adopt the element-wise product and have $\boldsymbol{x}_i \odot \boldsymbol{x}_j$ and $\boldsymbol{w}_i \odot \boldsymbol{w}_j$. To extract deep features, we concatenate all the raw deep representation as the output of this layer:

$$\boldsymbol{c} = \boldsymbol{s}_i \oplus \boldsymbol{s}_j \oplus \boldsymbol{p}_i \oplus \boldsymbol{p}_j. \tag{5}$$

After features fusion, we inject the users' interest influence into the *Neural Trust Network* from both deep and shallow parts.

Multilayer Perceptron Layer. To model the complex relationships of user-user pairs, we feed the raw deep representation $\boldsymbol{c}$ into a fully connected feed-forward neural network. Afterward, MLPs further learn the high-level deep features. As MLP repeats layer by layer, for layer $l = 1, \ldots, L$, we combine them to get the output:

$$\boldsymbol{h}_L = g_{L-1}(\ldots g_1({\mathrm{Z}_1}^T \cdot \boldsymbol{c}) + \boldsymbol{b}_1 \ldots) + \boldsymbol{b}_{L-1}, \tag{6}$$

where Z_x and $\boldsymbol{b}_x$ denote the weight matrix and bias vector for the x^{th} layer's perceptron, respectively. The output $\boldsymbol{h}_L$ aggregates informative social interactions and represents the high-level deep features of interactions.

Output Layer. To predict the final score $\hat{t}_{ij}$, we incorporate high-level deep feature with fused shallow MF representations as:

$$\boldsymbol{h}_f = [\boldsymbol{h}_L, \boldsymbol{x}_i \odot \boldsymbol{x}_j, \boldsymbol{w}_i \odot \boldsymbol{w}_j]. \tag{7}$$

After that, an extra full-connected layer will be used to transfer $\boldsymbol{h}_f$ to a predictive score $\hat{t}_{ij}$.

$$\hat{t}_{ij} = \sigma({\mathrm{Z}_L}^T \boldsymbol{h}_f + \boldsymbol{b}_L), \tag{8}$$

where σ means the widely used nonlinear function (Sigmoid, Relu, etc.).

Link-Based Loss Function. To better capture potential social relationships, we designed a link-based loss function, which could preserve the topology information, making the Neural Trust Network to have more ability of network inference. Specifically, we utilize the shortest path length among social users and define the loss function as:

$$Y_{ij} = \begin{cases} 0, & if \quad \omega = 0, d > \omega, \ or \ d = \infty \\ \omega + 1 - d, & if \quad \omega + 1 - d > 0, \end{cases} \tag{9}$$

$$L_s = -\sum_{(i,j,t_{ij}) \in T^+ \cup T^-} \left(\frac{Y_{ij}}{\omega} - \hat{t}_{ij} \right)^2, \tag{10}$$

where ω is the receptive field of path length, which limits the farthest distance we focus and decides the length of the social relationship we care about. Y_{ij} is the relative similarity between user i and user j. Parameter d is the shortest distance between two social users in a graph. T^+ is the observed links set in $\mathbf{T}$, and T^- denotes the unobserved links, which are randomly sampled from missed unobserved feedbacks.

We argue that one of the advantages of this loss function is to preserve topology information, because the new link-based loss function not only directly models the loss of observed links, but also directly models the distance of two users in a social network. With a parameter ω, the proposed loss function could explicitly model up to the ω-th order structure of the social network. Specifically, when $\omega = 1$, the proposed loss function degenerates to classical link-based loss function that only models the first order structure. Therefore, our proposed new link loss-based function could alleviate the data sparsity issue in previous works. Here is an example—if user j is a positive sample of user i, they have the direct connection; thus, the d equals to 1. Thus, no matter what the value of ω is, the label given to the positive sample is always 1. For negative samples, the formula will make a difference between them, e.g., if we set $\omega = 2$, which means we care about those candidates who

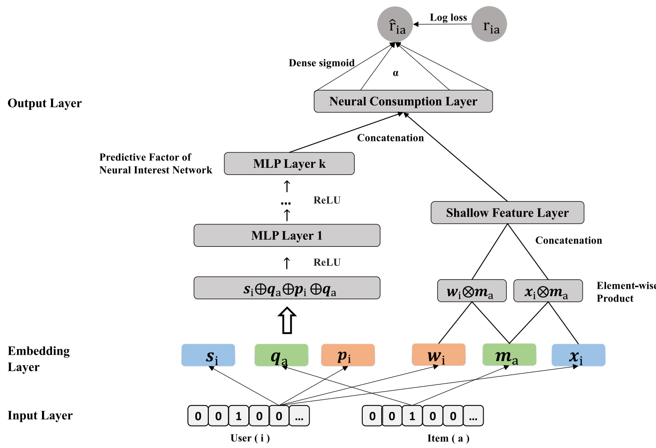


Fig. 4. The structure of the Neural Interest Network. We use the user features of different spaces to interact with the features of items, showing the impact of users on items in different parts, i.e., how users' social factor and preference factor itself impact the consumption preferences.

are relatively close to the current user in the graph (i.e., $d <= 2$). If a negative user is two steps far from the current user, then the final relative similarity is 0.5, which is distinguished from negative samples because they have a strong structural correlation in the graph, which means that it may be a potential social friend. With the increase of ω, the social members who are closer to each other will be strengthened. However, in some cases, it may be contrary to the real situation, since it is impossible for a user to know all of her/his closer nearby social members.

We argue that the above loss functions preserve the network topology from two folds: (1) The number of negative samples for each social user is similar to breadth traversal. The more negative samples, the larger scale we consider about. (2) The parameter ω controls the depth. By adjusting ω, we can adjust the length of the relationship we pay attention to on a graph. Therefore, ω and d control the depth traversal. Then, we are able to preserve the topological information by the designed link-based loss function.

4.2 Neural Interest Network

As mentioned above, *Neural Trust Network* is capable of modeling the social link behavior of users by combining the shallow MF model and the designed deep network. During this process, we considered the correlation of users' interest by feature fusion. Besides, we preserved localized topological information of each user on the prediction task. However, the consumption behavior of users is still unexplored. Thus, in this section, we detail the *Neural Interest Network* to model users' consumption behavior with the social impact, whose architecture is shown in Figure 4.

Input and Embedding Layer. Given a user-item pair (u_i, v_a), we utilize one-hot vectors to represent $u_i \in \mathbb{R}^{1\times|N|}$ and $v_a \in \mathbb{R}^{1\times|M|}$ as input. Then, we get the deep representations $\boldsymbol{s}_i$, $\boldsymbol{p}_i$, $\boldsymbol{q}_a$ by conducting the lookup operation from $\mathbf{S}$, $\mathbf{P}$ and the deep consumption item latent space $\mathbf{Q}$, respectively. To get shallow features, we perform the same operation on shallow latent spaces $\mathbf{X}$, $\mathbf{W}$, and the shallow consumption item latent space $\mathbf{M}$; then, we get: $\boldsymbol{x}_i$, $\boldsymbol{w}_i$, $\boldsymbol{m}_a$.

Fusion Layer. Similarly, we also consider two aspects: (1) Indeed, users' preferences may be influenced by the social network, i.e., $\boldsymbol{s}_i \oplus \boldsymbol{q}_a$, and the consumption behavior itself, i.e., $\boldsymbol{p}_i \oplus \boldsymbol{q}_a$. (2) We adopt element-wise product on shallow representations to measure the inner-product similarity [16], i.e., $\boldsymbol{x}_i \odot \boldsymbol{m}_a$ and $\boldsymbol{w}_i \odot \boldsymbol{m}_a$. To extract high-level deep features, we also concatenate all raw deep representations as the output of fusion layer, which is formulated as follows:

$$\boldsymbol{c}' = \boldsymbol{s}_i \oplus \boldsymbol{q}_a \oplus \boldsymbol{p}_i \oplus \boldsymbol{q}_a. \tag{11}$$

Multilayer Perceptron Layer. Due to the stacked structure of the MLP layer on the left channel, we adopt a similar strategy in the *Neural Trust Network*. Thus, for an L layers network, the output can be formulated as follows:

$$\boldsymbol{h}'_L = g_{L-1}(\ldots g_1(\mathbf{Z}_1'^T \cdot \boldsymbol{c}') + \boldsymbol{b}'_1 \ldots) + \boldsymbol{b}'_{L-1}. \tag{12}$$

Output Layer. At the last layer of MLP, we concatenate deep neural feature $\boldsymbol{h}'_L$ with the fused shallow MF features; then, we have:

$$\boldsymbol{h}'_f = [\boldsymbol{h}'_L, \boldsymbol{x}_i \odot \boldsymbol{m}_a, \boldsymbol{w}_i \odot \mathbf{m}_a], \tag{13}$$

where $\boldsymbol{h}'_f$ contains the high-level deep feature and shallow MF feature. Then, we output the predictive score $\hat{r}_{ia}$ by an extra fully-connected layer:

$$\hat{r}_{ia} = \sigma\left(\mathbf{Z}_L'^T \boldsymbol{h}'_f + \boldsymbol{b}'_L\right). \tag{14}$$

Loss Function. We adopted the pair-wised logloss, which has been widely used for recommendation models with good performance [4, 58]; then, we have:

$$L_c = -\sum_{(i,a,r_{ia}) \in R^+ \cup R^-} (r_{ia} log(\hat{r}_{ia}) + (1 - r_{ia}) log(1 - \hat{r}_{ia})), \tag{15}$$

where R^+ denotes the set of positive samples (observed user-item pairs), and R^- denotes the set of negative samples (unobserved user-item pairs that randomly sampled from $\mathbf{R}$).

4.3 Model Training

In general speaking, we focus on both *social link behavior* and *consumption behavior* of users, which are similar to the widely used point-wise in Probabilistic Matrix Factorization (PMF) [37] and pair-wised ranking-based loss function in Bayesian personalized ranking (BPR) [41]; we also design a loss function for optimization:

$$\begin{aligned} \min L_{s+c} = \lambda_s & \sum_{(i,j,t_{ij}) \in T^+ \cup T^-} \left(\frac{Y_{ij}}{\omega} - \hat{t}_{ij}\right)^2 \\ - \lambda_c & \sum_{(i,a,r_{ia}) \in R^+ \cup R^-} (r_{ia} log(\hat{r}_{ia}) + (1 - r_{ia}) log(1 - \hat{r}_{ia})) + \theta, \end{aligned} \tag{16}$$

where θ is the regularization parameters in NJBP. We initialize the training parameters with a Gaussian distribution with a mean value of 0 and a variance of 0.01. All the parameters in the above loss function are differentiable.

In practice, we implement the proposed model with TensorFlow[1] framework, and we train the model parameters by Stochastic Gradient Descent (we adopt Adam optimizer) with mini-batch, and we set the learning rate as 0.0005. Since we have different numbers of the training set on two kinds of behaviors data, how to divide the training set into batches is also important, as it may have different results. One approach is to fix the same batch-size for two prediction tasks. However, the part with more training samples will have more batches, and, hence, will draw more attention in the resulting model, making this part overfitting. Therefore, we split the mini-batch by fixing the number of batches, i.e., two prediction tasks have the same number of batches with an individual batch-size. Specifically, we set batch size 1,000 for consumption behavior prediction, and then we calculate the same batches for social link behavior prediction task.

[1] https://www.tensorflow.org/.

ALGORITHM 1: Joint training algorithm for NJBP model

Input: The rating matrix **R**, social network matrix **T**, and the shortest distance table **D**;
Output: The ranked lists of prediction results on two tasks.
Initialize all the model parameters set θ;
while Not converged **do**
  for All users $i \in U$ **do**
    for Each user-item(i, a) pair and each social user-user(i, j) pair from user i **do**
      Calculate the output $\hat{r}_{ia}$ (Equation (14)).
      Look up the shortest distance d_{ij} in $\boldsymbol{D}$;
      Calculate the ouput $\hat{t}_{ij}$ (Equation (8)).
    end for
    Update all parameters in the objective function (Equation (16)).
  end for
end while

In our designed loss function, λ_s and λ_c are the trade-off parameters; they play an important role in the final results. FIP [59] claimed that friendship information is helpful for consumption preference modeling. However, too large weight on friendship will later pollute the interest prediction performance. Therefore, we will study the impact in the next section. Moreover, if we set $\lambda_s = 0$, it becomes NeuMF.

The training algorithm of NJBP is shown in Algorithm 1. Besides, for the structure of neural networks in our model, we adopt the pyramid structure with 3-layer from the bottom to the top; the dimension of each layer decreases by half.

5 EXPERIMENTS

In this section, we conduct experiments with answering the following four key questions, which aim at verifying the effectiveness of our proposed methods:

RQ1: How does NJBP perform compared to other state-of-the-art single task models?
RQ2: Is the designed link-based loss function useful for providing more accurate results?
RQ3: Does the feature fusion impose influence on the performance of the NJBP model?
RQ4: How do the key parameters impact the performance of our NJBP model?

5.1 Experimental Settings

Dataset Description. We conduct experiments on two real-world publicly available datasets: Epinions[2] [42] and Flickr[3] [19]. The Epinions is publicly available[4] and the Flickr dataset comes from HASC [50]; it is downloadable on online.[5] They are all who-trusts-whom online social network platforms. For both datasets, we retain the users who are relatively active (the boundary value of active users is flexible), which means we will delete the users or items whose appearance times are less than the set value. Besides, the original ratings are presented with detailed rating values, as we use implicit feedback [41] on consumption behavior prediction, we transform the original ratings to binary values. Specifically, for all rating data on two datasets, if a rating score is greater than or equal to 3, we mark it as 1; otherwise, it equals 0.

[2] http://www.epinions.com/.
[3] https://www.flickr.com.
[4] www.librec.net/datasets.html.
[5] https://github.com/newlei/HASC.

Table 2. The Statistics of the Data after Splitting

Dataset	Epinions	Flickr
Users	7,186	5,259
Items	12,953	20,333
Training Consumption	808,425	742,330
Test Consumption	38,105	20,276
Validation Consumption	4,962	55,358
Training Links	1,062,885	447,060
Test Links	51,503	36,006
Validation Links	6,956	6,649
Consumption Sparsity	99.83%	99.87%
Link Sparsity	99.59%	99.68%

Then, we split datasets into training set, test set, and validation set. For both behaviors that are contained in Epinions and Flickr, we randomly select 20% of the record of each user as the test set to evaluate the performance. After that, 5% of the record of each user is randomly selected as the validation set to tune the parameter and decide the stopping of the training model. The remaining record is the training set to optimize the designed loss function. An overview of the detail characteristics of two datasets after pruning is summarized in Table 2. As we only observe limited positive feedbacks of users, we follow the same operation in many works [16, 51] and randomly sample r times (we set the default sample ratio $r = 4$) missing unobserved feedbacks as pseudo negative feedbacks at each iteration in the training process, and we reselect the pseudo negative samples in each iteration. Therefore, each observed interaction will be randomly paired with four (flexible parameter) times the negative samples. Thus, the number of training instances is much more than the number of data in the test set and validation set.

Evaluation Metrics. When we evaluate the performance, it is impossible to consider all users and items. Thus, we randomly pick up 100 negative samples for each user (i.e., 100 negative users for social link behavior prediction and 100 negative items for consumption behavior prediction; all these negative samples have had no interactions with the current user before). Then, we mix the negative samples and corresponding positive samples for each user in the test set together. After that, we input the mixed test set into the trained model, which outputs the rating scores for each user. The top-K scored candidates will be recommended to users as potential social friends and preferred items. In this way, we alleviate the time-consuming problem of ranking all users and items for each user during evaluation.

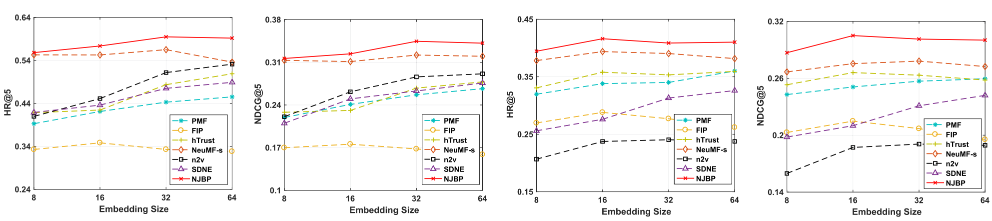
In terms of evaluation metrics, we adopt two widely used metrics in recommender systems, *Hit Ratio (HR@K)* [11] and *Normalized Discounted Cumulative Gain (NDCG@K)* [14]. *HR@K* can be interpreted as the percentage of users successfully recommended, and *NDCG@K* is a precision-based measure that accounts for the relative accuracy predicted position of the positive instances. The larger the value, the better the performance for both of the two metrics. In the experiments, we set $K = 5$ for both metrics, and we report the average results of all methods on the test set.

5.2 Performance Comparison (RQ1)

We compare the performance of NJBP with other methods. For better illustration, we summarize the details of these models in Table 3, which clearly shows the differences between our method and others. All parameters in the baselines are carefully tuned to ensure the best performance and fair comparison. For neural network based models (NJBP, NeuMF, NNCF), the embedding size of the

Table 3. Characteristics of the Baselines, with C and S Denoting the Consumption Behavior and Social Link Behavior, Respectively

Model	Data Source		Prediction?		Model Types	
	C	*S*	*C*	*S*	*Shallow Model*	*Deep Model*
PMF [37]	×	√	×	√	√	×
SocialMF [20]	√	√	√	×	√	×
BPR [41]	√	×	√	×	√	×
hTrust [46]	√	√	×	√	√	×
Node2vec [12]	×	√	×	√	√	×
SDNE [47]	×	√	×	√	×	√
NeuMF-s [16]	×	√	×	√	√	√
NeuMF-c [16]	√	×	√	×	√	√
NNCF [4]	√	√	√	×	√	√
FIP [59]	√	√	√	√	√	×
NJBP	√	√	√	√	√	√



(a) HR@5 on Epinions (b) NDCG@5 on Epinions (c) HR@5 on Flickr (d) NDCG@5 on Flickr

Fig. 5. Performance of social link behavior prediction on two datasets. The Embedding Size of abscissa means the dimensions of users and items for shallow model and the dimensions of last layer for deep neural network based models.

MLP predictive factor (last layer of MLP) plays the role of controlling their modeling capability [16]. For shallow MF models, the embedding size of latent representations plays the same role. Therefore, we explore the performance of different embedding size (*Embedding Size* with 8, 16, 32, and 64). All results pass the two-tailed paired t-test with p-value < 0.05.

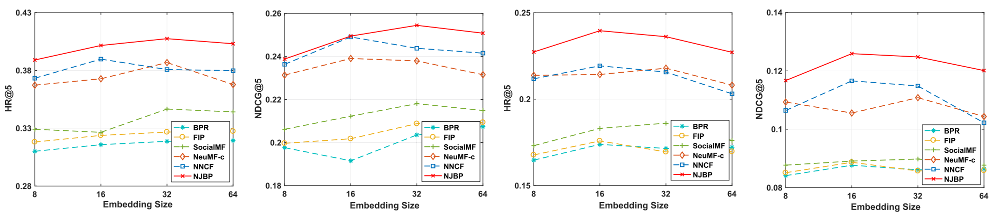
5.2.1 Social Link Behavior Prediction Performance. We report the performance of NJBP model with the following social link behavior prediction methods; the results are shown in Figure 5.

PMF [37]. Strictly speaking, PMF is proposed for consumption behavior prediction. It is a traditional shallow MF method, which is modeled by reducing the loss between the reconstructed rating matrix and the original rating matrix $\boldsymbol{R}$. As our method distinguishes negative sampled users, we adopt PMF on the social link behavior prediction task for fair comparison.

FIP [59]. FIP is presented for both social link behavior prediction and consumption behavior prediction. Specifically, FIP shares the same user representations.

hTrust [46]. hTrust utilizes the consumption preference to improve the performance of trust prediction; it adopts two pieces of behavioral information.

Node2vec [12]. Node2vec is a state-of-the-art network embedding method, which improves the random walk strategy on the basis of Deepwalk [39]; and Node2vec is very suitable for social link behavior prediction.



(a) HR@5 on Epinions (b) NDCG@5 on Epinions (c) HR@5 on Flickr (d) NDCG@5 on Flickr

Fig. 6. Performance of consumption behavior prediction on two datasets. The Embedding Size of abscissa means the dimensions of users and items for shallow model and the dimensions of last layer for deep neural network based models.

SDNE [47]. Structural Deep Network Embedding (SDNE) follows the idea of an autoencoder; it utilizes neighborhood information as input to preserve local topology structure. SDNE can be seen as a deep neural network based graph embedding method.

NeuMF-s [16]. We follow the idea of NeuMF to predict the potential relationships in a social network, it combines the shallow MF feature and the deep neural network feature.

On social link behavior prediction, when embedding size is 64, the relative improvement over NeuMF-s is 5.1% on social link behavior prediction. As the social network matrix **T** is sparse, SDNE is not able to balance various approximation well. The hTrust performs better than FIP and node2vec on social link behavior prediction. It shows the effectiveness of leveraging the homophily effect. Besides, even PMF can be seen as a special case of FIP; FIP still underperforms PMF and other methods on two datasets. We guess the possible reasons are that our datasets lack of the profiles of users and attributes of items, thus, some prior information, has not been concluded. We also argue that another shortcoming of FIP comes from the shared user latent representations, which need to fit two kinds of behaviors, making the performance discounted. For NJBP, it embraces two advantages. First, NJBP can identify potential relationships from a graph. Besides, for user representations, NJBP has two different latent spaces (social latent space and consumption latent space) for two prediction tasks to avoid feature confusion. Therefore, NJBP provides the best performance in most cases, achieving significant improvements over other methods.

5.2.2 Consumption Behavior Prediction Performance. We compare consumption prediction results with the following methods; the results are shown in Figure 6.

BPR [41]. BPR optimizes the traditional matrix factorization (MF) with a pair-wise ranking loss. We employ it to perform the consumption behavior prediction as it is a highly competitive baseline for the ranking task.

FIP [59]. FIP is presented for both social link behavior prediction and consumption behavior prediction. Specifically, FIP shares the same user representations.

SocialMF [20]. SocialMF incorporates the social neighborhood information with the users' preference to perform consumption behavior prediction.

NeuMF-c [16]. NeuMF is a SOTA method on recommendation task, which combines a traditional shallow MF model and deep neural networks into a framework. By adopting deep features from deep neural networks, NeuMF can fit more complex interactions between users and items.

NNCF [4]. NNCF improves the performance of NeuMF by utilizing the neighbor information of users and items. Therefore, NNCF is a social recommendation model with deep neural networks.

On consumption behavior prediction, NJBP outperforms NeuMF-c and NNCF with about 5.4% and 7.1% on default embedding size 64 without any pre-training. NNCF is not better than NeuMF when the embedding size is 64, there is no obvious winner between them. By adopting multiple

Table 4. The Impact of the Parameter ω

(a) The performance of social link behavior prediction with different ω.

	Epinions		Flickr	
	HR	NDCG	HR	NDCG
$\omega = 1$	0.585	0.334	0.402	0.291
$\omega = 2$	0.604	0.349	0.423	0.308
$\omega = 3$	0.599	0.342	0.410	0.299
$\omega = 4$	0.574	0.319	0.426	0.309

(b) The performance of consumption behavior prediction with different ω.

	Epinions		Flickr	
	HR	NDCG	HR	NDCG
$\omega = 1$	0.400	0.256	0.237	0.126
$\omega = 2$	0.416	0.269	0.239	0.128
$\omega = 3$	0.410	0.264	0.237	0.124
$\omega = 4$	0.389	0.244	0.247	0.133

neural networks, these deep models have a more powerful ability to fit more complex interactions, and they usually outperform shallow MF models. Moreover, we can see, even with a large dimension size 64, shallow MF models substantially underperform deep neural network models with a small embedding size on the last layer of MLP. Among all shallow MF models, SocialMF incorporates neighborhood information with traditional MF model; thus, it is better than BPR and FIP. FIP shares the same user latent representations, and due to missing user profiles and item attributes in both datasets, FIP does not perform well.

5.3 The Effectiveness of Link-Based Loss Function (RQ2)

As stated before, NJBP identifies potential social relationships by adopting a link-based loss function. To demonstrate the effectiveness of it, we tune the ω in the range of $[1, 2, 3, 4]$ to observe the results. We don't set a larger value (maximum four) because two nodes that far apart from each other in a graph are rarely linked, e.g., we may never know a friend of our friend's friend.

We utilize the shortest path length between two nodes to measure the similarity of relationships, and we discard the relationships among users with long path distance ($d > \omega$) or there is no path between two users. The results have been shown in Table 4(a) and (b). If we set $\omega = 1$, it turns into the implicit feedback with the pair-wise loss function. We attempt to a larger value $\omega = 2$; the performance increases a lot on both prediction tasks. And when we set $\omega = 3$, the performance starts to decline, though it is still better than the beginning. Then, we continue to increase the value of ω, and the performance becomes worse. We conclude that a large ω will confuse positive and negative samples, as many negative samples become pseudo positive samples, making the result worse. Thus, the performance increases at first and then decreases, indicating most of the social relationships are direct or indirect with only one extra hop in a social graph. This conclusion is consistent with a recent GCN-based model where just two GCN structure layers get the best performance [54, 60], which shows that considering the neighborhood relationships of two steps in a graph possibly make the information utilization more accurate. In addition, another interesting finding is that the quality of consumption behavior prediction will be affected by the social part; we analyze this phenomenon in Section 5.5. Moreover, the performance improvement also shows the effectiveness of the use of n-hop information and the defectiveness of extracting negative samples randomly. For the same reason, we also conduct the same operation on NeuMF-s, PMF, and FIP; they show similar performance improvement and no more descriptions here to avoid duplication.

In order to further verify the effectiveness of the link-based loss function, we show a case study of social link behavior prediction tasks on Epinions. Specifically, we list the output scores between a user and corresponding positive and negative samples; the results of different methods are shown in Table 5. In this test, we randomly select a user #146 and observe her social link behavior predic-

Table 5. A Case Study Results of User #146

	(+)#421/2	(+)#157/3	(+)#1327/2	(-)#242/2
NJBP	0.868	0.782	0.766	0.743
NeuMF – s	0.826	0.806	0.720	0.708
SDNE	0.646	0.588	0.670	0.622
Node2vec	0.640	0.668	0.736	0.578
hTrust	0.668	0.600	0.813	0.448
PMF	0.648	0.642	0.626	0.602
FIP	0.554	0.540	0.532	0.501

Table 6. Characteristics of *NJBP* and Five Variants

Model	Social Link Behavior Prediction		Consumption Behavior Prediction	
	Deep Latent Space	Shallow Latent Space	Deep Latent Space	Shallow Latent Space
	S P Q	X W M	S P Q	X W M
NJBP	√ √ ×	√ √ ×	√ √ √	√ √ √
JDN	√ √ √	× × ×	√ √ √	× × ×
JSN	× × ×	√ √ √	× × ×	√ √ √
NJIN	√ √ ×	√ √ ×	× √ √	× √ √
NJTN	√ × ×	√ × ×	√ √ √	√ √ √
NMBP	√ × ×	√ × ×	× √ √	× √ √

tion result of NJBP. The top four user IDs are #421, #157, #1327 and, #242 (with descending order of scores, corresponding to the second row in the table). For other methods, we also calculate the scores with these four users (non-descending order) to observe the prediction results.

We use (+) to represent positive samples while (-) represents the negative sample; the shortest path length is displayed beside the user #ID. In this example, we note that NJBP captures most of the positive samples (get a higher score); it has significantly higher accuracy compared with other methods (lower scores mean that in other methods, these four users may not be ranked in the top recommendation position). From the table, we also find that user #242 is the unique negative sample. However, predicted by our method, user #242 is also strongly correlated with user #146 (high predicted score 0.743). The reason is that user #242 is only a 2-hop length from user #146 in the social network graph, so user #242 is captured by our proposed link-based loss function.

Also, in order to better understand the distance among users, we calculate the shortest path length of all pairs (user-user) in two datasets; then we get the average length: Epinions (2.188) and Flickr (2.845). It reveals the majority of user relationships do not extend too far in a social network. Because the data is taken from the real world, we believe this phenomenon has a strong realistic effect.

5.4 Ablation Study (RQ3)

NJBP utilizes both deep and shallow latent spaces; for each of them, there are two user latent spaces and an item latent space. To demonstrate the effectiveness of feature fusion in NJBP, we compare the performance with five variants in Table 6.

Joint Deep Network (*JDN*) and *Joint Shallow Network* (*JSN*) mean that we only utilize deep representations and shallow MF representations, respectively. By comparing their performance, we can verify the effectiveness of the combination of deep and shallow features. Another three

Table 7. Performance of *NJBP* and Five Variants

Model	Social Link Behavior				Consumption Behavior			
	Epinions		Flickr		Epinions		Flickr	
	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
NJBP	**0.594**	**0.344**	**0.408**	**0.300**	**0.407**	**0.254**	**0.250**	**0.129**
JDN	0.482	0.288	0.364	0.279	0.381	0.242	0.195	0.106
JSN	0.467	0.260	0.351	0.258	0.338	0.217	0.201	0.104
NJIN	0.570	0.323	0.389	0.286	0.389	0.240	0.218	0.115
NJTN	0.553	0.311	0.366	0.271	0.374	0.233	0.238	0.123
NMBP	0.526	0.293	0.358	0.262	0.360	0.225	0.175	0.097

Table 8. The Impact of Negative Sampling Ratio r

(a) Performance of social link behavior prediction with different sampling ratio r.

	Epinions		Flickr	
	HR	NDCG	HR	NDCG
r = 1	0.551	0.304	0.385	0.280
r = 2	0.584	0.332	0.403	0.295
r = 3	0.590	0.338	0.409	0.298
r = 4	0.604	0.347	0.421	0.306
r = 5	0.614	0.354	0.429	0.313
r = 6	0.624	0.360	0.415	0.304
r = 7	0.618	0.357	0.411	0.300
r = 8	0.612	0.351	0.405	0.298

(b) Performance of consumption behavior prediction with different sampling ratio r.

	Epinions		Flickr	
	HR	NDCG	HR	NDCG
r = 1	0.408	0.260	0.219	0.111
r = 2	0.402	0.257	0.236	0.124
r = 3	0.410	0.263	0.234	0.123
r = 4	0.415	0.266	0.237	0.124
r = 5	0.423	0.272	0.240	0.124
r = 6	0.427	0.274	0.245	0.131
r = 7	0.425	0.273	0.246	0.130
r = 8	0.420	0.269	0.240	0.126

models reveal the impact of feature fusion. In *Neural Joint Interests Network* (*NJIN*), we discard the deep and shallow social user latent space (**S** and **X**) in Neural Interest Network. Similarly, in *Neural Joint Trust Network* (*NJTN*), we don't consider consumption user latent spaces (**P** and **W**) when predicting social link behavior. In *Neural Multi Behavior Prediction* (*NMBP*), users' two latent spaces don't have any feature fusions. We list the results in Table 7, in which NJBP get the best performance because of encompassing the advantages of other variants.

5.5 Parameter Analysis (RQ4)

Sampling Ratio. The advantage of the flexible negative sampling has been shown in many works [41, 44]. In this section, we flexibly control the sampling ratio r to explore its influence. We empirically fix $r = 4$ on social part/consumption part while we debug the consumption part/social part. Then, we show the performance of NJBP w.r.t. different negative sampling ratio (range from 1 to 8) on two tasks in Table 8(a) and (b).

With the increase of sampling ratio, the performance of social link behavior prediction first increases and then decreases. It shows that too large sampling ratio may adversely hurt the performance [16]. The optimal sampling ratio is around 4 to 6, and the performance of the consumption behavior part has the same tendency with the social part.

The Number of Layers. In many cases, properly increasing the layers will improve performance. However, as far as we know, there are few works that study the influence of the number of

Table 9. HR@5 of MLP with Different Layers on Flickr

Task	Social Link Behavior				Consumption Behavior			
Layers / Factor d	MLP-1	MLP-2	MLP-3	MLP-4	MLP-1	MLP-2	MLP-3	MLP-4
d = 8	0.419	0.393	0.406	0.379	0.213	0.227	0.226	0.241
d = 16	0.382	0.396	0.419	0.419	0.209	0.231	0.237	0.242
d = 32	0.409	0.388	0.425	0.390	0.231	0.224	0.248	0.252
d = 64	0.416	0.405	0.387	0.374	0.235	0.226	0.255	0.260

Table 10. NDCG@5 of MLP with Different Layers on Flickr

Task	Social Link Behavior				Consumption Behavior			
Layers / Factor d	MLP-1	MLP-2	MLP-3	MLP-4	MLP-1	MLP-2	MLP-3	MLP-4
d = 8	0.305	0.285	0.296	0.278	0.111	0.118	0.121	0.125
d = 16	0.276	0.290	0.306	0.305	0.109	0.121	0.125	0.122
d = 32	0.300	0.283	0.312	0.285	0.121	0.115	0.131	0.139
d = 64	0.312	0.295	0.284	0.272	0.126	0.118	0.134	0.146

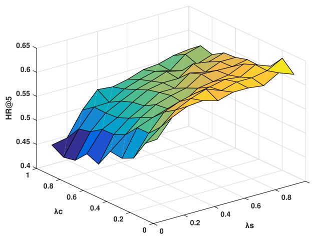
layers on a joint neural network model. Therefore, it is meaningful to observe whether a deeper network structure is able to improve performance.

We set the default number of layers to three. When we change the number of the network layer on the social/consumption part, the consumption/social part is fixed. Moreover, we conduct the experiment with different embedding sizes on the last MLP layer. Tables 9 and 10 have shown the results on Flickr; the evaluation on Epinions shows a similar performance.

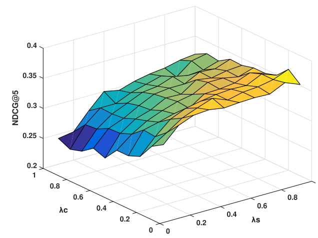
As we can see, MLP-3 gets the best performance on social link behavior prediction in some cases, nevertheless, it is not stable. Some other work on single behavior prediction has come to the same conclusion [58]. On consumption behavior prediction task, the results are contrary to the previous one; it is consistent with the conclusion in [16] that both deeper network and larger dimensions can improve the performance of the model to some extent. We guess a possible reason is that in both *Neural Interest Network* and *Neural Trust Network*, we have adopted the same network structure. However, the social network matrix **T** has a higher density than the rating matrix **R**. Therefore, on the task of social link behavior prediction, the model is more prone to overfitting. Therefore, a feasible and direct solution is to appropriately reduce the number of layers or the dimensions of the embedding size of Multilayer Perceptron(MLP) in *Neural Trust Network*, making it perform better on the test set.

The Impact of Balance Parameter. NJBP consists of Neural Trust Network and Neural Interest Network, and there are two important balance parameters: λ_s and λ_c in Equation (16). They balance the L_s and L_c. In order to explore the impact of them, we fix one of the parameters and vary the other for evaluation. The values of the two parameters is range from 0.1 to 1 with the interval of 0.1. We show the intuitive results in Figures 7 and 8. On Epinions, the result of social link behavior prediction shows an upward trend when the λ_s is relatively larger and λ_c takes a moderate value. It indicates the powerful social influence. For consumption behavior, excluding extreme conditions (one of the parameters is mostly 0.1, and the other is large), the performances slightly increase and float.

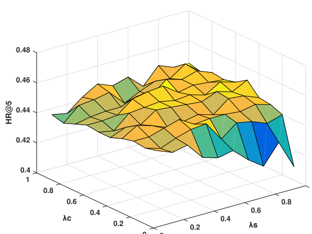
On Flickr, the results of both behaviors become better as λ_s increases, verifying the effective influence of social impact on consumption behavior. In fact, the results on two datasets are not the



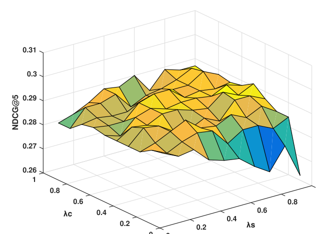
(a) HR@5 of social link behavior



(b) NDCG@5 of social link behavior

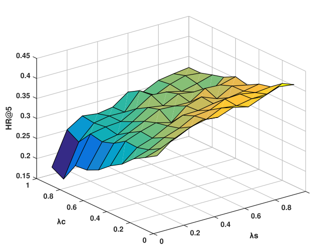


(c) HR@5 of consumption behavior

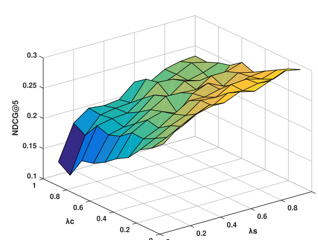


(d) NDCG@5 of consumption behavior

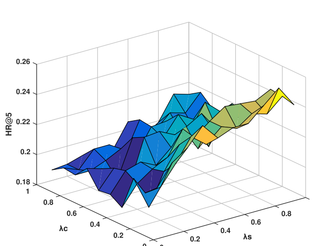
Fig. 7. The parameter analysis for NJBP on Epinions. With the increase of λ_s, the performance of social link behavior prediction shows an upward trend, while the effect of consumption behavior prediction slightly rises and floats.



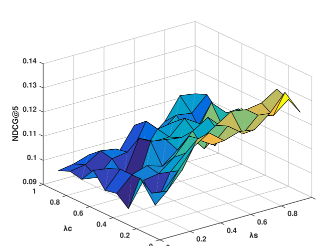
(a) HR@5 of social link behavior



(b) NDCG@5 of social link behavior



(c) HR@5 of consumption behavior



(d) NDCG@5 of consumption behavior

Fig. 8. The parameter analysis for NJBP on Flickr. With the increase of λ_s, the performance of social link behavior prediction shows an upward trend, and it boosts the performance of consumption behavior prediction.

first time for discovering the phenomenon of social influence. FIP exploited homophily to establish an integrated network linking a user to interested services and connecting different users with common interests [59]. In the joint user behavior learning, the social factor may be useful for stimulating consumption behavior, as social influence is very common in our daily life.

6 CONCLUSION

In this article, we presented a new unified deep neural based framework named Neural Joint Behavior Prediction Model (NJBP) for users' two kinds of behaviors prediction, i.e., *consumption behavior* and *social link behavior*. Specifically, we designed the *Neural Interest Network* for consumption behavior and the *Neural Trust Network* for social link behavior, and we modeled the correlations between them by the fusion layers in both of them. Besides, we designed a link-based loss function in the *Neural Trust Network* to preserve the topological information in a social network. After that, we designed a joint optimization function to combine the correlation between these two behaviors for better prediction. Finally, we conducted a comprehensive set of experiments on two real-world datasets, and the corresponding experimental results demonstrated that NJBP outperforms other state-of-the-art approaches on the task of top-N user behaviors prediction. In the future, as items usually contain rich attributes (e.g., image, text, category), we will improve NJBP by adopting these multimedia and auxiliary information, not only for the purpose of improving the accuracy of prediction but also providing clearer interpretable results of users' behavior prediction.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 6 (2005), 734–749. DOI: https://doi.org/10.1109/TKDE.2005.99

[2] Charu Aggarwal and Karthik Subbian. 2014. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)* 47, 1 (2014), 10. DOI: https://doi.org/10.1145/2601412

[3] Sinan Aral, Lev Muchnik, and Arun Sundararajan. 2009. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences* 106, 51 (2009), 21544–21549. DOI: https://doi.org/10.1073/pnas.0908800106

[4] Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1979–1982. DOI: https://doi.org/10.1145/3132847.3133083

[5] Ghazaleh Beigi, Jiliang Tang, Suhang Wang, and Huan Liu. 2016. Exploiting emotional information for trust/distrust prediction. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 81–89. DOI: https://doi.org/10.1137/1.9781611974348.10

[6] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2018. Graph convolutional matrix completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[7] Xuezhi Cao, Haokun Chen, Xuejian Wang, Weinan Zhang, and Yong Yu. 2018. Neural link prediction over aligned networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 249–256.

[8] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. 2015. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 119–128. DOI: https://doi.org/10.1145/2783258.2783296

[9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10. DOI: https://doi.org/10.1145/2988450.2988454

[10] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* (2018), 833–852. DOI: https://doi.org/10.1109/TKDE.2018.2849727

[11] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Transactions on Information Systems* 22, 1 (2004), 143–177. DOI: https://doi.org/10.1145/963770.963776

[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864. DOI: https://doi.org/10.1145/2939672.2939754

[13] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731. DOI: https://doi.org/10.24963/ijcai.2017/239

[14] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. 2015. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 1661–1670. DOI: https://doi.org/10.1145/2806416.2806504

[15] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 355–364. DOI: https://doi.org/10.1145/3077136.3080777

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182. DOI: https://doi.org/10.1145/3038912.3052569

[17] Dirk Helbing, Dirk Brockmann, Thomas Chadefaux, Karsten Donnay, Ulf Blanke, Olivia Woolley-Meza, Mehdi Moussaid, Anders Johansson, Jens Krause, Sebastian Schutte, et al. 2015. Saving human lives: What complexity science and information systems can contribute. *Journal of Statistical Physics* 158, 3 (2015), 735–781. DOI: https://doi.org/10.1007/s10955-014-1024-9

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366. DOI: https://doi.org/10.1016/0893-6080(89)90020-8

[19] Mark J. Huiskes and Michael S. Lew. 2008. The MIR Flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*. ACM, 39–43. DOI: https://doi.org/10.1145/1460096.1460104

[20] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems*. ACM, 135–142. DOI: https://doi.org/10.1145/1864708.1864736

[21] Glen Jeh and Jennifer Widom. 2002. SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 538–543. DOI: https://doi.org/10.1145/775047.775126

[22] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2789–2802. DOI: https://doi.org/10.1109/TKDE.2014.2300487

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[24] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*. ACM, 208–211. DOI: https://doi.org/10.1145/1352793.1352837

[25] Elizabeth A. Leicht, Petter Holme, and Mark E. J. Newman. 2006. Vertex similarity in networks. *Physical Review E* 73, 2 (2006), 026120. DOI: https://doi.org/10.1103/PhysRevE.73.026120

[26] Xiaoyi Li, Nan Du, Hui Li, Kang Li, Jing Gao, and Aidong Zhang. 2014. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 289–297. DOI: https://doi.org/10.1137/1.9781611973440.33

[27] Defu Lian, Yuyang Ye, Wenya Zhu, Qi Liu, Xing Xie, and Hui Xiong. 2016. Mutual reinforcement of academic performance prediction and library book recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1023–1028. DOI: https://doi.org/10.1109/ICDM.2016.0130

[28] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1754–1763. DOI: https://doi.org/10.1145/3219819.3220023

[29] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2018. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering* 30, 12 (2018), 2257–2270. DOI: https://doi.org/10.1109/TKDE.2018.2819980

[30] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031. DOI: https://doi.org/10.1002/asi.20591

[31] Dekang Lin et al. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Vol. 98. Citeseer, 296–304.

[32] Qi Liu, Han Wu, Yuyang Ye, Hongke Zhao, Chuanren Liu, and Dongfang Du. 2018. Patent litigation prediction: A convolutional tensor factorization approach. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 5052–5059. DOI: https://doi.org/10.24963/ijcai.2018/701

[33] Sanda Martinčić-Ipšić, Edvin Močibob, and Matjaž Perc. 2017. Link prediction on Twitter. *PloS One* 12, 7 (2017), e0181079. DOI: https://doi.org/10.1371/journal.pone.0181079

[34] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444. DOI: https://doi.org/10.1146/annurev.soc.27.1.415

[35] Aditya Krishna Menon and Charles Elkan. 2011. Link prediction via matrix factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 437–452. DOI: https://doi.org/10.1007/978-3-642-23783-6_28

[36] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*. 1045–1048.

[37] Andriy Mnih and Ruslan R. Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. 1257–1264.

[38] Matjaž Perc, Mahmut Ozer, and Janja Hojnik. 2019. Social and juristic challenges of artificial intelligence. *Palgrave Communications* 5, 1 (2019), 1–7. DOI: https://doi.org/10.1057/s41599-019-0278-x

[39] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710. DOI: https://doi.org/10.1145/2623330.2623732

[40] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000. DOI: https://doi.org/10.1109/ICDM.2010.127

[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[42] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. 2003. Trust management for the semantic web. In *International Semantic Web Conference*. Springer, 351–368. DOI: https://doi.org/10.1007/978-3-540-39718-2_23

[43] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, 285–295.

[44] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 926–934. http://papers.nips.cc/paper/5028-reasoning-with-neural-tensor-networks-for-knowledge-base-completion.pdf.

[45] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 185–194. DOI: https://doi.org/10.1145/3209978.3210023

[46] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Exploiting homophily effect for trust prediction. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*. ACM, 53–62. DOI: https://doi.org/10.1145/2433396.2433405

[47] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1225–1234. DOI: https://doi.org/10.1145/2939672.2939753

[48] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. (2019), 417–426. DOI: https://doi.org/10.1145/3331184.3331267

[49] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*. ACM, 495–503. DOI: https://doi.org/10.1145/3018661.3018689

[50] Le Wu, Lei Chen, Richang Hong, Yanjie Fu, Xing Xie, and Meng Wang. 2019. A hierarchical attention model for social contextual image recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2019). DOI: https://doi.org/10.1109/TKDE.2019.2913394

[51] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. 2017. Modeling the evolution of users' preferences and social links in social networking services. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1240–1253. DOI: https://doi.org/10.1109/TKDE.2017.2663422

[52] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Bai Long, and Zhenya Huang. 2016. Modeling users' preferences and social links in social networking services: A joint-evolving perspective. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 279–286.

[53] Le Wu, Qi Liu, Enhong Chen, Nicholas Jing Yuan, Guangming Guo, and Xing Xie. 2016. Relevance meets coverage: A unified framework to generate diversified recommendations. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 39. DOI: https://doi.org/10.1145/2700496

[54] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. DOI: https://doi.org/10.1145/3331184.3331214

[55] Le Wu, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2018. Collaborative neural social recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2018). DOI: https://doi.org/10.1109/TSMC.2018.2872842

[56] Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, Yanjie Fu, and Meng Wang. 2020. Joint item recommendation and attribute inference: An adaptive graph convolutional network approach. *arXiv preprint arXiv:2005.12021* (2020). DOI: https://doi.org/10.1145/3397271.3401144

[57] Peizhi Wu, Yi Tu, Xiaojie Yuan, Adam Jatowt, and Zhenglu Yang. 2018. Neural framework for joint evolution modeling of user feedback and social links in dynamic social networks. In *Proceedings of the 27th International Joint Conference of Artificial Intelligence*. 1632–1638. DOI: https://doi.org/10.24963/ijcai.2018/226

[58] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3203–3209. DOI: https://doi.org/10.24963/ijcai.2017/447

[59] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. 2011. Like like alike: Joint friendship and interest propagation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, 537–546.

[60] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 974–983. DOI: https://doi.org/10.1145/3219819.3219890

[61] Shuangfei Zhai and Zhongfei Zhang. 2015. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 451–459. DOI: https://doi.org/10.1137/1.9781611974010.51

[62] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 5. DOI: https://doi.org/10.1145/3285029

Received June 2019; revised December 2019; accepted June 2020