Neural Framework for Joint Evolution Modeling of User Feedback and Social Links in Dynamic Social Networks

Peizhi Wu[*1]**, Yi Tu**[2]**, Xiaojie Yuan**[1]**, Adam Jatowt**[3]**, Zhenglu Yang**[*1]

[1] CCCE, Nankai University, Tianjin, China
[2] George Washington University, Washington, DC, USA
[3] Kyoto University, Kyoto, Japan
wupz@mail.nankai.edu.cn, niklaus_yi@gwu.edu, yuanxj@nankai.edu.cn,
adam@dl.kuis.kyoto-u.ac.jp, yangzl@nankai.edu.cn

Abstract

Modeling the evolution of user feedback and social links in dynamic social networks is of considerable significance, because it is the basis of many applications, including recommendation systems and user behavior analyses. Most of the existing methods in this area model user behaviors separately and consider only certain aspects of this problem, such as dynamic preferences of users, dynamic attributes of items, evolutions of social networks, and their partial integration. This work proposes a comprehensive general neural framework with several optimal strategies to jointly model the evolution of user feedback and social links. The framework considers the dynamic user preferences, dynamic item attributes, and time-dependent social links in time-evolving social networks. Experimental results conducted on two real-world datasets demonstrate that our proposed model performs remarkably better than state-of-the-art methods.

1 Introduction

Over the last years, social network services (SNSs) have become one of the most important and popular tools for users to communicate, share, and deliver information. The available information from SNSs (e.g., user attributes and preferences) and related resources (e.g., user purchase logs) allow in-depth understanding and modeling of user behaviors, thereby potentially benefiting certain important applications, such as recommendation systems, link prediction, and so forth.

In SNSs, giving feedback about items consumed by users and creating social links with others are two pivotal types of user behaviors that have been individually studied by many previous works [Mnih and Salakhutdinov, 2008; Salakhutdinov *et al.*, 2007; Liben-Nowell and Kleinberg, 2007]. However, social scientists have demonstrated that the two core behaviors in SNSs, i.e., user feedback and social links, are not

---

[*]Corresponding author.

independent of each other [Crandall *et al.*, 2008]. Specifically, future user preferences are affected by their social influences, such as *word of mouth*. Moreover, users who share similar consumption preferences have a high likelihood to associate with one another. This phenomenon is called the homophily effect. To address these observations, most of the prior studies utilized only one of the two behaviors to boost the prediction of another behavior [Jamali and Ester, 2010; Tang *et al.*, 2013]. Only a few works attempted the joint modeling of the two user behaviors, which enhances the predictions of user behaviors [Wu *et al.*, 2016; Jamali *et al.*, 2011].

In the real world, social networks evolve over time, and user preferences are likewise dynamic. In addition, item attributes often change as well, as a function of user inclinations and other factors [Wu *et al.*, 2017a]. These time-evolving aspects make the effective and comprehensive modeling of user behavior increasingly challenging. Most previous works jointly modeled user feedback and social link behaviors, with the assumption of static social networks, or considered only a partial integration of the three dynamic aspects [Yang *et al.*, 2011; Wu *et al.*, 2017b]. To address these issues, we propose to jointly model the temporal evolution of user feedback and social link behaviors via capturing the dynamic user preferences, dynamic item attributes and time-evolving social networks. We introduce a social network embedding framework to explore how social influences and the homphily effect impact future user feedback and social linking behaviors.

Recently, neural networks (NN) have achieved remarkable successes in computer vision, speech recognition, and text processing. NNs have been then applied by many studies to model user feedback or social links and have shown promising results. However, the number of NN-based approaches that jointly model the two user behaviors and benefit both tasks of user feedback and social link prediction, is limited.

In this paper, we present a novel NN-based model to jointly learn the two user behaviors by considering the three aforementioned dynamic aspects in SNS. The main contributions of this paper are as follows:

- We propose Neural Joint Modeling (NJM), a novel deep learning based architecture to model the evolution of user feedback and social links in dynamic social networks by

considering dynamic user preferences, dynamic item attributes, and time-evolving social networks. This work is the first one based on deep learning that jointly models the evolution of the two user behaviors.

- We leverage expressive NNs to thoroughly study the high-level nonlinear interactions between user historical preference and social influence as well as between user social structure and the homophily effect.

- We conduct extensive experiments on two real-world datasets, and the experimental results illustrate that our model achieves a promising effectiveness in recommendation and social link prediction compared with state-of-the-art methods.

2 Related Work

We first summarize the general approaches for separately modeling user feedback and predicting social links. Then we introduce the joint modeling of these two user behaviors.

User feedback is mainly affected by user preferences and item attributes, which can be modeled by Collaborative Filtering (CF). One of the most traditional CF techniques is matrix factorization (MF) [Mnih and Salakhutdinov, 2008]. Since user preferences and item attributes generally change over time certain MF models consider temporal dynamics by creating handcrafted features [Koren, 2010; Xiong et al., 2010]. However, with such an approach, they cannot capture complex temporal patterns [Wu et al., 2017a]. Recently, deep learning-based techniques have been widely leveraged for MF in modeling user feedback in static networks [He et al., 2017; Zheng et al., 2016]. By contrast, [Wu et al., 2017a] proposed recurrent recommender network that can model the dynamics of users and items. Based on the social influence theory, many studies integrated the context of user friendship with feedback data in static social networks [Jamali and Ester, 2010; Deng et al., 2017].

The problem of link prediction viewed as computing the node proximityis to predict potential new links on the basis of partially observed connections in a social network. Early approaches predicted possible links only in static networks [Liben-Nowell and Kleinberg, 2007]. With the availability of temporal information in social networks, other scholars analyzed the patterns of the evolution of networks [Zhu et al., 2016; Sarkar et al., 2012; Yu et al., 2017]. A few deep learning based models have been used for the link prediction problem recently. For example, [Kipf and Welling, 2016] devised variational graph autoencoder to learn meaningful latent embeddings on the link prediction task. [Liao et al., 2017] leveraged attributed social network embedding for link prediction. [Zhang and Chen, 2017] proposed a next-generation link prediction method, called Weisfeiler-Lehman Neural Machine. To utilize the homophily effect, [Tang et al., 2013] proposed to exploit user preferences on items for link prediction.

[Jamali et al., 2011] analyzed the temporal dynamics of social networks by using bidirectional effects, but that work cannot be used for personalized recommendation. [Yang et al., 2011] jointly modeled the two user behaviors by using a unified framework that assumes a static representation of

social networks. [Wu et al., 2017b] proposed a model that captures the temporal dynamics of the two user behaviors yet it assumes item attributes to be static. Nevertheless, no existing work has explored the idea of taking into account all the three aforementioned dynamic aspects to jointly learn the two kinds of user behaviors.

To the best of our knowledge, our model is the first one based on deep learning that jointly models the evolution of the two user behaviors and considers dynamic item attributes, dynamic user preferences, and time-dependent social links.

3 The Proposed Model

3.1 Problem Definition

A social network has a set of users $U = \{u_1, u_2, ..., u_N\}$ and a set of items $V = \{v_1, v_2, ..., v_M\}$. Users consume items and create links with others over time. The feedback data on items expressed by users at time t are described in a feedback matrix $\boldsymbol{C}^t \in \mathbb{R}^{N \times M}$. $\boldsymbol{C}_{a,i}^t$ denotes the rating of user u_a on item v_i at time t. On the other hand, social links built by users are given in a link matrix $\boldsymbol{S}^t \in \mathbb{R}^{N \times N}$. $\boldsymbol{S}_{a,b}^t = 1$ means that user u_a forms a link with user u_b at time t. We use a, b to represent users and i, j to denote items.

Given two matrix sequences: a feedback matrix sequence $C = [\boldsymbol{C}^1, \boldsymbol{C}^2, ..., \boldsymbol{C}^T]$ and a social link matrix sequence $S = [\boldsymbol{S}^1, \boldsymbol{S}^2, ..., \boldsymbol{S}^T]$, the tasks of our model are to: (1) quantify the social influence in user feedback behavior and homophily effect in social link behavior; (2) depict the dynamic states of items; and (3) predict each user's new feedback and social link behaviors at the next time.

3.2 Overall Framework

To solve the tasks mentioned above, we propose a framework that comprises two modules: (1) user feedback model (Fig. 1, left) which captures user dynamic preferences by quantifying historical user preferences and social influences, and which uses LSTM to learn the dynamic states of items; (2) social link model (Fig. 1, right) which considers the homophily effect when predicting social links. The two modules are jointly trained to achieve superior prediction performance.

Given the dimension of the latent space as D, we propose two time-dependent embeddings (1) $\boldsymbol{u}_a^t$, which is called *latent preference embedding* and describes the latent preference of user u_a on items at time t and (2) $\boldsymbol{p}_a^t$, which is called *latent link embedding* and represents the latent structure of user u_a at time t. Moreover, we denote $\boldsymbol{U}^t = \{\boldsymbol{u}_a^t\} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{P}^t = \{\boldsymbol{p}_a^t\} \in \mathbb{R}^{N \times D}$. The one-hot vector is employed for user ID, as shown at the bottom of Fig. 1.

3.3 User Feedback Model

User feedback is determined by user preferences and item attributes. In following, we introduce the user preference and item attribute parts of this model.

User Preference

The new preferences of each user are affected by their historical preferences and social influences.

Impact Factors of User Preference. $\boldsymbol{u}_a^t$ represents the time-dependent latent preference of user u_a at time t. On the
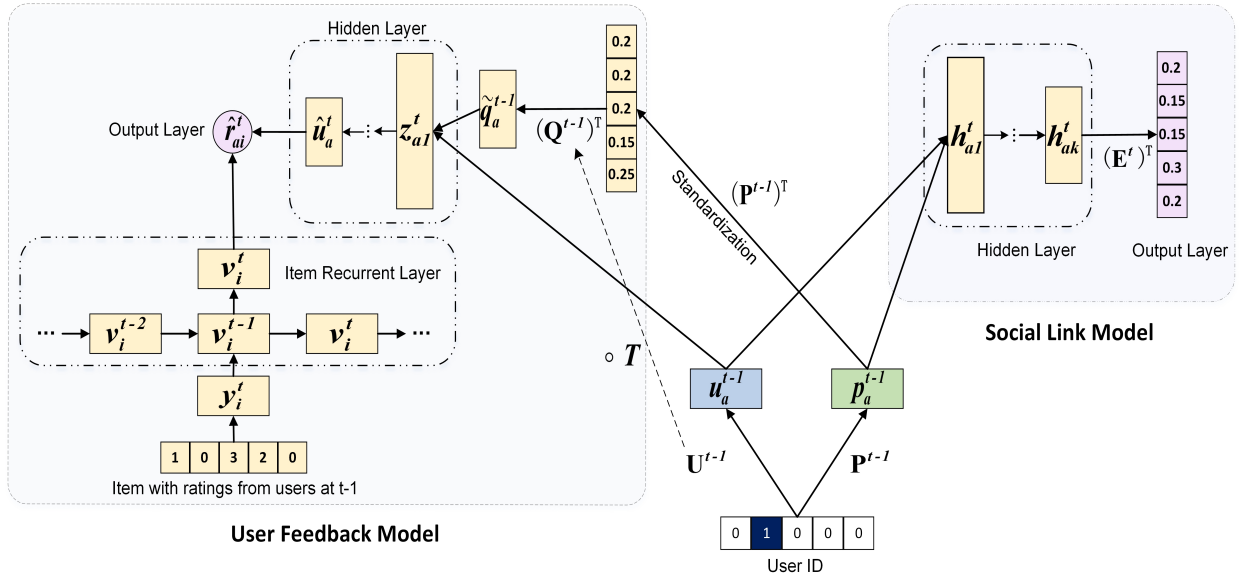
Figure 1: The structure of the proposed model

other hand, we propose $\boldsymbol{q}_a^t$, which is called *latent influence embedding* and describes the time-dependent latent influence of user u_a on other users at time t. Consequently, we have $\boldsymbol{Q}^t = \{\boldsymbol{q}_a^t\} \in \mathbb{R}^{N \times D}$.

In most previous works [Wu *et al.*, 2017b], time-dependent latent preference and time-dependent latent influence were considered equal. In our work, we argue that both terms do not describe the same concept because users may exaggerate or diminish their specific tastes and characters due to some reasons. For example, an introverted user may perform actively in SNSs as he might want to appear outgoing; meanwhile, another user may hide his preferences on some features, such as crime movies, to protect his privacy. Therefore, we propose an effective strategy to correlate $\boldsymbol{u}_a^t$ and $\boldsymbol{q}_a^t$ here.

We use the recommendation transformation factor matrix, denoted as $\boldsymbol{T} \in \mathbb{R}^{N \times D}$, to describe the change from users' preferences to users' influences on other users. Each row in $\boldsymbol{T}$ reflects the extent to which the corresponding user exaggerates or diminishes each feature. Furthermore, users' trust degrees on others vary. Therefore, we propose $\boldsymbol{f}_a^{t-1}$ to denote the vector of the trust score of user u_a on his friends at time *t-1*. Specifically, the b-th element in $\boldsymbol{f}_a^{t-1}$ is user u_a's trust score in user u_b at time $t-1$. Besides, let $\mathcal{N}_a^t$ denote the set of linked users of u_a till time t and $\boldsymbol{I}_a^t$ denote the friend indicator vector. $\boldsymbol{I}_{ab}^t$ equals 1 if $b \in \mathcal{N}_a^t$, otherwise it equals 0. Here, one rational way to set $\boldsymbol{f}_a^{t-1}$ is to use the node proximity between the user and his friends: $\boldsymbol{I}_a^{t-1} \circ (\boldsymbol{p}_a^{t-1}(\boldsymbol{P}^{t-1})^T)$, where $\circ$ denotes the Hadamard product operation. In addition, we need to standardize the product to ensure the trust scores sum of 1. Thus, we employ $\tilde{\boldsymbol{q}}_a^{t-1}$ to represent the time-dependent latent influence of friendship on user u_a:

$$
\begin{aligned}
\tilde{\boldsymbol{q}}_a^{t-1} &= \boldsymbol{f}_a^{t-1}\boldsymbol{Q}^{t-1} = \boldsymbol{f}_a^{t-1}(\boldsymbol{U}^{t-1} \circ \boldsymbol{T}) \\
&= \tau(\boldsymbol{I}_a^{t-1} \circ (\boldsymbol{p}_a^{t-1}(\boldsymbol{P}^{t-1})^T))(\boldsymbol{U}^{t-1} \circ \boldsymbol{T}),
\end{aligned} \quad (1)
$$

where τ represents the standardization function.

Integrating Impact Factors of User Preference. To predict the new preferences of user u_a at time t, we need to integrate

the user preference embedding $\boldsymbol{u}_a^{t-1}$ and the latent influence of friendship on user u_a, $\tilde{\boldsymbol{q}}_a^{t-1}$. In [Wu *et al.*, 2017b], the author implemented this idea by linear combination. Inspired by [He *et al.*, 2017], we believe that the simple linear combination of $\boldsymbol{u}_a^{t-1}$ and $\tilde{\boldsymbol{q}}_a^{t-1}$ will limit the representation capability of the model.

A standard Multi-Layer Perceptron (MLP) is adopted in our model to identify the nonlinear interaction between $\boldsymbol{u}_a^{t-1}$ and $\tilde{\boldsymbol{q}}_a^{t-1}$. The output of the user preference hidden layer is the latent preference prediction vector of user a at time t, $\hat{\boldsymbol{u}}_a^t$. And we obtain $\hat{\boldsymbol{U}}^t = \{\hat{\boldsymbol{u}}_a^t\} \in \mathbb{R}^{N \times D}$.

The MLP model under the user feedback model is defined as:

$$
\begin{aligned}
\boldsymbol{z}_{a1}^t &= \phi_1^f(\boldsymbol{u}_a^{t-1}, \tilde{\boldsymbol{q}}_a^{t-1}), \\
\boldsymbol{z}_{ai}^t &= \phi_i^f(\boldsymbol{W}_i\boldsymbol{z}_{i-1}^t + \boldsymbol{b}_i), i = 2, ..., n_f, \\
\hat{\boldsymbol{u}}_a^t &= \boldsymbol{z}_{an_f}^t,
\end{aligned} \quad (2)
$$

where n_f denotes the number of feedback hidden layers, and $\boldsymbol{W}_L$, $\boldsymbol{b}_L$ denote the weight matrix and bias vector. No social influence is present at time 1; thus, we set $\boldsymbol{z}_{a1}^1 = \phi_1^f(\boldsymbol{u}_a^1, \boldsymbol{0})$.

Item Attribute

In our model, we suppose that the item attributes change over time (e.g., a restaurant serves different food in different seasons). We utilize the LSTM recurrent NNs to determine the inherent item dynamics instead of the states of items. This technique can help address the vanishing gradient problem and enables us to incorporate past observations and predict future trajectories in an integrated manner. We use $\boldsymbol{v}_i^t$ to denote the latent attributes embedding of item v_i at time t. In addition, we use a nonparametric model to learn the inherent item dynamics. Then, we can extrapolate the future behaviors of the items. Here, our model learns only the dynamics of items and not their states.

The parameters of an item are dependent on its popularity and the users who have consumed it. Accordingly, the in-

put of the item attribute model are items with ratings from users at a previous time. Then, we use a transformation matrix to learn how to project this information into an embedding space, and obtain the item latent vector $\boldsymbol{y}_i^t$, which serves as the input to the LSTM. The output of the item attribute part $\boldsymbol{v}_i^t$ is the latent attribute embedding of item v_i at time t: $\boldsymbol{v}_i^t = LSTM(\boldsymbol{v}_i^{t-1}, \boldsymbol{y}_i^t)$.

Objective

The output of the user preference part and the item attribute part serves as the input of the recommendation output layer. The states of user u_a and item v_i change with time, but we assume that some components of them still remain stationary, such as the item genre or user gender. Thus, we propose the dynamic vectors $\hat{\boldsymbol{u}}_a^t$ and $\boldsymbol{v}_i^t$ with stationary components $\boldsymbol{u}_a$ and $\boldsymbol{v}_i$: $\hat{r}_{ai}^t = sigmoid(< \hat{\boldsymbol{u}}_a^t, \bar{\boldsymbol{v}}_i^t > + < \boldsymbol{u}_a, \boldsymbol{v}_i >)$, where $<,>$ denotes the inner product operation and $\bar{\boldsymbol{v}}_i^t$ is the affine function of $\boldsymbol{v}_i^t$. The loss function of user feedback model is defined as follows:

$$L_u = \sum_{(a,i,t)\in train} (r_{ai}^t - \hat{r}_{ai}^t)^2. \quad (3)$$

3.4 Social Link Model

The construction of a social link is mainly determined by two factors. First, users who share similar consumption preferences have a high probability to form a link connection. This phenomenon is called homophily effect. Second, users who have high node proximity are highly likely to be associated with one other and make friends.

Hence, given the latent link embeddings and latent preference embeddings, this model aims to estimate the time-dependent pairwise link probability of nodes. We use the softmax function to define the conditional link probability of user u_b on u_a at time t:

$$p^t(\mathrm{u}_b|\mathrm{u}_a) = \frac{exp(g(\boldsymbol{u}_a^{t-1}, \boldsymbol{u}_b^{t-1}, \boldsymbol{p}_a^{t-1}, \boldsymbol{p}_b^{t-1}))}{\sum_{b'=1}^{N} exp(g(\boldsymbol{u}_a^{t-1}, \boldsymbol{u}_{b'}^{t-1}, \boldsymbol{p}_a^{t-1}, \boldsymbol{p}_{b'}^{t-1}))}, \quad (4)$$

where g is the function that map nodes u_a and u_b to their estimated link score.

Furthermore, we maximize the likelihood function for the global social link modeling:

$$l = \prod_{t=1}^{T} \prod_{a=1}^{N} \prod_{b\in\mathcal{N}_a^t} p^t(\mathrm{u}_b|\mathrm{u}_a). \quad (5)$$

When predicting user u_a's link probability at time t, an intuitive method to combine the homophily effect and node proximity involves first measuring the similarity of user preference and user link embeddings between user u_a and others directly and then integrating the two similarities via linear addition [Wu et al., 2017b].

$$\hat{\boldsymbol{s}}_a^t = \sigma(\boldsymbol{u}_a^{t-1}(\boldsymbol{U}^{t-1})^T, \boldsymbol{p}_a^{t-1}(\boldsymbol{P}^{t-1})^T), \quad (6)$$

where $\hat{\boldsymbol{s}}_a^t$ is the link probability vector of u_a at time t, σ is the linear addition function.

However, this simple method ignores the nonlinear relationship among the heterogeneous factors. In the social

link model, we explore the nonlinear interaction between the homophily effect and node proximity, which lets a deep NN to learn the complex interactions between the two factors.

Input Layer. Two components $\boldsymbol{u}_a^{t-1}$ and $\boldsymbol{p}_a^{t-1}$ serve as the input of this layer. One component captures the latent information of the user preference, and the other component represents the latent structure of user u_a.

Hidden Layer. After feeding $\boldsymbol{u}_a^{t-1}$ and $\boldsymbol{p}_a^{t-1}$ to the multi-layer NN, we obtain $\boldsymbol{h}_{ak}^t$ from the output of the last hidden layer. $\boldsymbol{h}_{ak}^t$ represents the abstract embedding, which bridges the preference information and social structure information of user u_a.

$$\begin{aligned}\boldsymbol{h}_{a1}^t &= \phi_1^s(\boldsymbol{u}_a^{t-1}, \boldsymbol{p}_a^{t-1}), \\ \boldsymbol{h}_{ak}^t &= \phi_k^s(\boldsymbol{W}_k \boldsymbol{h}_{a(k-1)}^t + \boldsymbol{b}_k), k = 2, ..., n_s,\end{aligned} \quad (7)$$

where n_s represents the number of link hidden layers and ϕ^s denotes the activation function. The homophily effect is absent at time 1, so we set $\boldsymbol{h}_{a1}^1 = \phi_1^s(\boldsymbol{0}, \boldsymbol{p}_a^1)$.

Output Layer. In this layer, the output of the last hidden layer is transformed into a probability vector $\hat{\boldsymbol{s}}_a^t$, which is the link probability of u_a at time t.

We denote the time-dependent user neighborhood embedding of user u_a at time t as $\boldsymbol{e}_a^t$ and assume that $\boldsymbol{e}_a^t$ is evolved from user abstract embedding $\boldsymbol{h}_{ak}^t$. In addition, We denote $\boldsymbol{E}^t = \{\boldsymbol{e}_a^t\} \in \mathbb{R}^{N\times D}$. Therefore, the regularization of this evolution is defined as:

$$L_{r_e} = \sum_{t=1}^{T} \sum_{a=1}^{N} \|1 - \boldsymbol{e}_a^t(\boldsymbol{h}_{ak}^t)^T\|_F^2 + \|1 - \boldsymbol{e}_a^{t+1}(\boldsymbol{e}_a^t)^T\|_F^2. \quad (8)$$

To be detailed, we obtain

$$g(\boldsymbol{u}_a^{t-1}, \boldsymbol{u}_b^{t-1}, \boldsymbol{p}_a^{t-1}, \boldsymbol{p}_b^{t-1}) = < \boldsymbol{e}_b^t, \boldsymbol{h}_{ak}^t >, \quad (9)$$

which can be fed into Eq. 4.

Predicting all the missing links in social networks is computationally inefficient. To address this issue, we apply negative sampling procedure [Jamali et al., 2011]. Thus, we minimize the cross-entropy loss function defined as

$$L_s = - \sum_{(a,b,t)\in Z^+} log\hat{\boldsymbol{s}}_{ab}^t - \sum_{(a,b,t)\in Z^-} log(1 - \hat{\boldsymbol{s}}_{ab}^t), \quad (10)$$

where $Z^+ \subset Z$ denotes the set of positive link interactions in Z, and $Z^- \subset Z$ denotes the set of negative interactions in Z.

3.5 Joint Training

To penalize large deviations of user embedding over time, we obtain

$$L_r = L_{r_e} + \sum_{t=2}^{T} \sum_{a=1}^{N} \|1 - \hat{\boldsymbol{U}}_a^t(\boldsymbol{U}_a^t)^T\|_F^2 + \|1 - \boldsymbol{P}_a^t(\boldsymbol{P}_a^{t-1})^T\|_F^2. \quad (11)$$

The whole loss function of our framework is defined as

$$L = L_u + \lambda L_s + \beta L_r + R(\theta), \quad (12)$$

Algorithm 1 Joint training of our model

Input: latent dimension D, the number of timestamps T, batch size B, the feedback sequence C, the social link sequence S

Output: All parameters in our framework θ that minimize the total loss L

1: **repeat**
2:   Sample a batch of $(\mathrm{u}_a, \mathrm{v}_i, C_{ai} = \{\boldsymbol{C}_{ai}^1, ..., \boldsymbol{C}_{ai}^T\}$, $C_{*i} = \{\boldsymbol{C}_{*i}^1, ..., \boldsymbol{C}_{*i}^T\}, \mathcal{N}_a = \{\mathcal{N}_a^1, ..., \mathcal{N}_a^T\}, Z_a \subset Z)$ of size B
3:   **for** $t \leftarrow 1$ to T **do**
4:     Accumulate feedback prediction loss L_u by Eq. 3
5:     Accumulate link prediction loss L_s by Eq. 10
6:     Accumulate evolution smoothness loss L_r by Eq. 11
7:   **end for**
8:   Accumulate the total loss L by Eq. 12
9:   Take a gradient step to optimize L
10: **until** L converges or is sufficiently small
11: **return** θ

Dataset	users	items	feedback records	links
Gowalla	21,755	71,139	330,602	257,550
Epinions	4630	26,991	65,683	78,356

Table 1: The statistics of the benchmark datasets

Figure 2: Performance comparison of user feedback prediction on RMSE

where λ is the trade-off parameter between user feedback prediction loss and social link prediction loss. β regularizes the evolution of user embeddings over time.

We implement our neural framework with TensorFlow to jointly train the multiple optimization objectives in Eq. 12 by performing stochastic gradient descent with mini-batch Adam. Algorithm 1 illustrates the SGD-based joint training procedure of our model.

Our model has two variants. If we do not minimize the social link prediction loss, then the degenerated model NJM-F can be used to make direct user feedback prediction. If we disregard the optimization of user feedback prediction loss, then we can use the degenerated model NJM-L to make a direct social link prediction.

4 Experiments

In this section, we describe the two-fold experiments for feedback prediction and link prediction. The source code is available at `https://github.com/NJMCODE2018/NJM`.

4.1 Experimental Setting

We evaluate our model on two benchmark datasets, i.e., the who-trust-whom online product sharing dataset *Epinions* and the location based social networking dataset *Gowalla*. In the data preparation procedure, we filter out users with less than two feedback records and two social link records. Table 1 shows the basic statics of the two datasets after filtering. During data splitting, we use the data up to time T as the training data and the new behaviors in $T+1$ as the test data. Without loss of generality, we normalize the user feedback ratings into the interval [0,1].

4.2 Feedback Prediction

Compared Algorithm. We compare our model with the following feedback prediction algorithms: *PMF* [Mnih and Salakhutdinov, 2008], *TimeSVD++* [Koren, 2010], *RRN* [Wu *et al.*, 2017a], *NCF* [He *et al.*, 2017], *FIP* [Yang *et al.*, 2011],
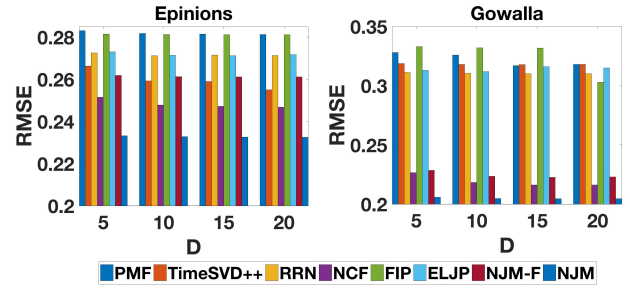
ELJP [Wu *et al.*, 2017b]. *FIP* and *ELJP* are joint training methods. We analyze all algorithms by measuring the accuracy of feedback prediction based on root mean-square error (RMSE) and we report the results with different numbers of dimensions. For fair comparison, we choose the best results of all algorithms to represent the best performances.

Performance Comparison. We observe from Figure 2 that NCF generates better results than MF based methods, and this comparison shows that applying deep learning methods on learning nonlinear interactions between users and items is effective. The experimental results show that our model achieves promising prediction effectiveness, yielding 7.4% and 9.1% improvements on Epinions and Gowalla data, respectively, over the best baselines. These results demonstrate that the neural structures of our model can accurately capture the dynamic user preferences with social influence and dynamic item attributes. Specifically, NJM performs better than NJM-F, which explains the benefits of jointly training the perspectives in feedback prediction.

4.3 Link Prediction

Compared Algorithm. Next, we quantitatively compare the proposed method with all the following link prediction algorithms: *AA* [Liben-Nowell and Kleinberg, 2007], *Node2Vec* [Grover and Leskovec, 2016], *SNE* [Liao *et al.*, 2017], *FIP* [Yang *et al.*, 2011], *ELJP* [Wu *et al.*, 2017b]. We use feedback records as the attributes of users and items in *SNE*. For convenient calculation, we follow the same procedure as one applied by [Wu *et al.*, 2017b] to process link prediction: for a test user u_a, we randomly sample 100 negative users that do not link to user u_a untill time t. Then, we mix these negative users and those positive linked users to select top K potential linked users as the link prediction results. We adopt three widely-used metrics: *precision@K*, *recall@K*, and *F1*. Here, we set K = 5 in the evaluation.

Performance Comparison. From the prediction results displayed in Figure 3, we observe that ELJP shows great performance. This proves that leveraging the homophily effect
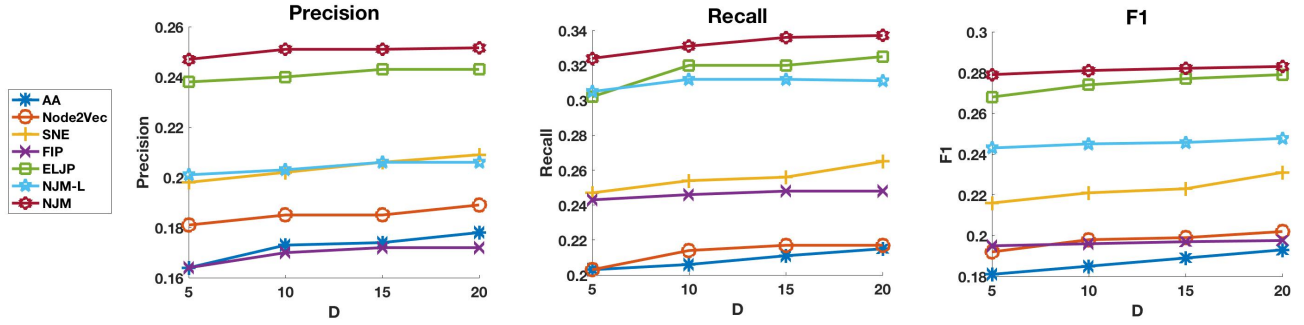
Figure 3: Performance comparison of link prediction on precision, recall and F1

User feedback model		Social link model			
n_f	RMSE	n_s	precision	recall	F1
0	0.269	0	0.203	0.298	0.237
1	0.236	1	0.246	0.318	0.278
2	0.233	2	0.249	0.327	0.282
3	0.231	3	0.249	0.330	0.284

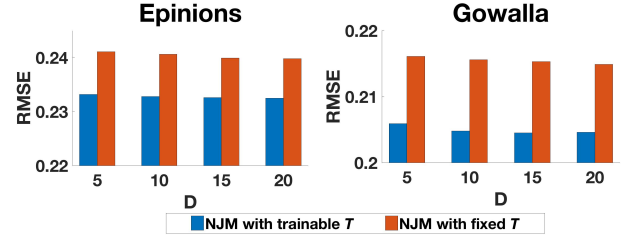Table 2: Performance of the proposed model using different number of hidden layers

Figure 4: Performance of the proposed model with the impact of the recommendation transformation factor matrix on RMSE

boosts social link prediction. Our model also has the best predictive power among all the proposed models on the Epinions dataset, which incorporates social network information and user feedback. This finding illustrates that exploring the high-level interactions between node proximity and the homophily effect is helpful. Moreover, our model exhibits better performance than NJM-P, and this result shows the power of jointly training for link prediction. The evaluation of different approaches on Gowalla shows similar performance and is not presented here due to space limitation.

4.4 Parameter Exploration

We set β 0.01 in Epinions data and 0.1 in Gowalla data. In addition, we explore how parameter differences influence the performance of our model through an in-depth analysis.

Neural Architectures. We apply different architectures of hidden layers in the user feedback and social link models in our framework to evaluate the model performance. We follow a tower pattern, where each successive layer has half the number of neurons of the lower layer. For example, if the number of latent dimensions is 10 and the number of hidden layers is 3, then the sizes of the layers are $40 \rightarrow 20 \rightarrow 10$. We evaluate the performance of the two models by varying the number of hidden layers from 0 to 3 with the number of latent dimensions D = 10. Table 2 demonstrates the prediction performance with different numbers of hidden layers in the two models. We find that 2 is the best number of hidden layers in both the models, considering the cost of training time and prediction results. Due to space limitations and the fact that the results on Gowalla data have similar performance, we only report the experimental results on Epinions data here.

Trade-off Parameter λ. λ is the trade-off parameter between user feedback prediction loss and social link prediction loss.

We set λ 0.1 in Epinions data and 1 in Gowalla data, in view of the balance of the two prediction tasks.

Recommendation Transformation Factor. We also investigate the effect of the recommendation transformation factor matrix T on feedback prediction. When the recommendation transformation factor is not considered, it means that T is fixed to be an all 1's matrix and untrainable. Figure 4 shows that the performance of NJM with trainable T is superior over the one with untrainable T on the two datasets. This demonstrates the usefulness of the proposed recommendation transformation factor matrix.

5 Conclusion

In this work, we propose a general end-to-end neural framework for modeling the evolution of user feedback and social links in dynamic social networks. We leverage sophisticated NNs to describe dynamic user preferences, dynamic item attributes, and time-evolving social links in a unified manner. We then obtain insights into the formation of the two user behaviors with social influence and the homophily effect using deep NNs. The experimental evaluation demonstrates the effectiveness of the introduced model.

People tend to form various communities according to their social relations, and these communities should have different contributions to the formation of user preferences, which is called the community effect. An interesting future work is the modeling of user feedback with social influence and the community effect. Our future work is thus to investigate this problem by expanding our model.

Acknowledgements

References

[Crandall *et al.*, 2008] David Crandall, Dan Cosley, Daniel Huttenlocher, Jon Kleinberg, and Siddharth Suri. Feedback effects between similarity and social influence in online communities. In *SIGKDD*, 2008.

[Deng *et al.*, 2017] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. On deep learning for trust-aware recommendations in social networks. *IEEE transactions on neural networks and learning systems*, 28(5):1164–1177, 2017.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, 2017.

[Jamali and Ester, 2010] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*, 2010.

[Jamali *et al.*, 2011] Mohsen Jamali, Gholamreza Haffari, and Martin Ester. Modeling the temporal dynamics of social rating networks using bidirectional effects of social relations and rating patterns. In *WWW*, 2011.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[Koren, 2010] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[Liao *et al.*, 2017] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *arXiv preprint arXiv:1705.04969*, 2017.

[Liben-Nowell and Kleinberg, 2007] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58(7):1019–1031, 2007.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2008.

[Salakhutdinov *et al.*, 2007] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML*, 2007.

[Sarkar *et al.*, 2012] Purnamrita Sarkar, Deepayan Chakrabarti, and Michael Jordan. Nonparametric link prediction in dynamic networks. *arXiv preprint arXiv:1206.6394*, 2012.

[Tang *et al.*, 2013] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *WSDM*, 2013.

[Wu *et al.*, 2016] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Bai Long, and Zhenya Huang. Modeling users' preferences and social links in social networking services: A joint-evolving perspective. In *AAAI*, 2016.

[Wu *et al.*, 2017a] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *WSDM*, 2017.

[Wu *et al.*, 2017b] Le Wu, Yong Ge, Qi Liu, Enhong Chen, Richang Hong, Junping Du, and Meng Wang. Modeling the evolution of users' preferences and social links in social networking services. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1240–1253, 2017.

[Xiong *et al.*, 2010] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, 2010.

[Yang *et al.*, 2011] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *WWW*, 2011.

[Yu *et al.*, 2017] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Haifeng Chen, and Wei Wang. Link prediction with spatial and temporal consistency in dynamic networks. In *IJCAI*, 2017.

[Zhang and Chen, 2017] Muhan Zhang and Yixin Chen. Weisfeiler-lehman neural machine for link prediction. In *SIGKDD*, 2017.

[Zheng *et al.*, 2016] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. A neural autoregressive approach to collaborative filtering. *arXiv preprint arXiv:1605.09477*, 2016.

[Zhu *et al.*, 2016] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2765–2777, 2016.