LINK PREDICTION WITH PERSONALIZED SOCIAL INFLUENCE

A Thesis

by

ZEPENG HUO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Xia Hu
Co-Chair of Committee,	James Caverlee
Committee Member,	Xiaoning Qian
Head of Department,	Dilma Da Silva

December 2017

Major Subject: Computer Engineering

ABSTRACT

Link prediction in social networks is to infer the new links likely to be formed next or to reconstruct the links that are currently missing. Link prediction is of great interest recently since one of the most important goals of social networks is to connect people, so that they can interact with their friends from real world or make new friend through Internet. So the predicted links in social networks can be helpful for people to have connections with each others. Other than the pure topological network structures, social networks also have rich information of social activities of each user, such as tweeting, retweeting, and replying activities.

Social science theories, such as social influence, suggests that the social activities could have potential impacts on the neighbors, and links in social networks are the results of the impacts taking place between different users. It motivates us to perform link prediction by taking advantage of the activity information.

There has been a lot of proposed methods to measure the social influence through user activity information. However, traditional methods assigned some social influence measures to users universally based on their social activities, such as number of retweets or mentions the users have. But the social influence of one user towards others may not always remain the same with respect to different neighbors, which demands a personalized learning schema. Moreover, learning social influence from heterogeneous social activities is a nontrivial problem, since the information carried in the social activities is implicit and sometimes even noisy.

Motivated by time-series analysis, we investigate the potential of modeling influence patterns based on pure timestamps, i.e., we aim to simplify the problem of processing heterogeneous social activities to a sequence of timestamps. Then we use timestamps

as an abstraction of each activity to calculate the reduction of uncertainty of one users social activities given the knowledge of another one. The key idea is that, if a user i has impact on another user j, then given the activity timestamps of user i, the uncertainty in user j's activity timestamps could be reduced. The uncertainty is measured by entropy in information theory, which is proven useful to detect the significant influence flow in time-series signals in information-theoretic applications.

By employing the proposed influence metric, we incorporate the social activity information into the network structure, and learn a unified low-dimensional representation for all users. Thus, we could perform link prediction effectively based on the learned representation. Through comprehensive experiments, we demonstrate that the proposed method can perform better than the state-of-the-art methods in different real-world link prediction tasks.

DEDICATION

In memory of my grandmother, Xianhong Yu.

ACKNOWLEDGMENTS

I wish to express my gratitude to my advisor, Dr. Xia Hu, for his continuing patience and guidance throughout my Master studies. I have acquired a lot from his passion and work ethic. The questions and ideas he brought up in our group meetings inspired me to come up with some novelties in my own research. I would also like to thank my advisory committee members: Dr. James Caverlee and Dr. Xiaoning Qian.

During two years at Texas A&M, I was fortunate to work with people in this cooperating team in DATA lab. I would like to thank them for be inspiring and creating. I gained a lot of help from my labmates, especially my labmate Xiao Huang, who contributed a lot in my research and helped in proofreading parts of the document.

Last but not least, I owe my deepest gratitude to my parents. Without their unfailing support and encouragement, this work would have not been possible.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Xia Hu as advisor, and Professor James Caverlee of the Department of Computer Science and Engineering, and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

My labmate Xiao Huang helped proofread the majority of the thesis document.

Funding Sources

This work was conducted without outside financial support.

NOMENCLATURE

SN	Social Networks
DSN	Directed Social Network
JC	Jaccard Coefficient
PSI	Personalized Social Influence
$\|\| \cdot \|\|_2$	the ℓ_2 norm of a matrix
$\|\| \cdot \|\|_F$	the Frobenius norm of a matrix
NS	Negative Sampling
BPR	Bayesian Personalized Ranking
AUC	Area Under the Curve
NLP	Natural Language Processing
MCM	Markov Chain Model
DI	Directed Information
LR	Logistic Regression
MDL	Minimum Description Length
KL	Kullback-Leibler
ASGD	Asynchronous Stochastic Gradient Descent
R&M	Retweet and Mention
CN	Common Neighbor
MF	Matrix Factorization
ELLR	Efficient Latent Link Recommendation

TI	Temporal Influence
SWR	Supervised Random Walk
MAP	Mean Average Precision
IC	Independent Cascade

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 The Overall Review of Link Prediction in Social Networks

With social networks becoming increasingly popular, predicting and reproducing the social network structure draws lots of attentions in recent years [1]. Among different problems in social networks, link prediction especially directed link prediction is of great interests [2], because in social networks users most often have directed links with each other, including in-links (followed by others) and out-links (following others). Link prediction is either to infer the links that are likely to occur in the near future or to reconstruct the existing links that are missing in the current snapshot of the social network.

1.2 Traditional Insights of Link Prediction in Social Networks

Due to its practical value, link prediction has become an effective computational tool for many real-world applications, such as friend recommendation [3], and community recommendation [4]. Traditional methods for link prediction can be roughly categorized into several groups. First, some methods use neighbor-based metrics to infer the missing links [5], where the similarity function can be the counts of common neighbors or some variations, such as Jaccard Coefficient of common neighbors. Second, some methods employ path-based metrics for link prediction, in which random walk is designed to traverse the paths between two users to calculate the proximity, with one hop or multiple hops [6]. As we can observe, existing work usually focused on the topological network structure for link prediction, while ignoring the fact that the links are actually from the social networks, where exists a rich set of social information of online users.

1.3 Motivation of Considering Social Influence

The various kinds of social activities carry abundant information of social media users, where one user's activities are complicatedly intertwined with other user's activities, through social influence [7] [8]. Social influence in social networks is defined as the phenomenon where we can observe "alteration of an attitude or behavior by one network actor in response to another" [9]. Therefore we can infer the existence of social influence by observing the changing pattern of social activities, which in turn means social influence is not equivalent to social activities, but some sort of quantification of social activities.

Social influence has been demonstrated useful for many applications, such as information diffusion study [10], and emotion contagion study [11], where social influence is quantified from social activities and thus has rich information of online users.

Although pure topological network structure has been intensively studied for link prediction, making use of both network structure and social influence remains an open problem. This motivates us to further investigate how to collectively model the two sources of information to improve the performance of link prediction in social networks.

1.4 The Challenges to Incorporate Social Influence to Link Prediction

However, it is non-trivial to integrate different types of social activities to infer social influence for link prediction. The challenges are as follows.

(1) The social activities are heterogeneous. In a social network like Twitter, users can tweet, retweet, reply, and mention others. The reasons for occurrences of various kinds of activities are different. Traditional methods tend to conduct prolonged feature selection processes on different social activities to extract useful information, which is not only time-consuming but also domain-specific, since each kind of activity needs a specifically designed feature selection algorithm.

(2) The manifestation of social influence is implicit. Users in social networks will

not explain who influences them or how they are influenced. Although we know users in social networks will receive social influence from people that they are following, it is implicit to quantify the influence she receives. According to social science theories, the following relationship is formed most likely after social influence has happened [12]. So the difficulty of inferring social influence will pose a challenge on predicting the links in social networks.

(3) The social influence of each user are neighbor-dependent, i.e. the social influence is not necessarily consistent with respect to different neighbors. Traditional social influence methods assigned the influence score exclusively to each user, i.e. one user only has one kind of influence score all the time, such as PageRank score [13] or Burt's network constraint score [14]. But that cannot represent the subtle difference when a user interacts with different neighbors. For example, in different scenarios of Pagerank, it has been shown that users have different directionality towards other users, i.e. one user can take the role as 'hub' with some users, but she may take the role as 'authority' with others. But universal social influence assignment can't preserve the subtle changes of personalized directionality in social networks, which, however, is the essential characteristic of directed link prediction.

1.5 High-level Idea of Proposed Method

Therefore, in this paper, we first simplify the complicated social activities into a time-series of timestamps as abstraction. And then we propose to use information-theoretic method to calculate the reduction of uncertainty of one user's activities given the knowledge of another one in a pair-wise manner, and use the entropy as the quantification of social influence. The learned social influence is later used as a regularization for a personalized learning framework, so that the link prediction will benefit from the personalized characteristics of each user. We call our method as PSI (Personalized Social Influence link

prediction).

We propose each user should take two representations in a pair-wise social influence schema, which are *Source* and *Target*[1], so that each user is no longer represented by a single unchanging influence score, but two vector representations carrying information of network structure and social influence. In this case, the user's directionality towards others can be preserved in two personalized representations individually, so that the subtle differences of one user's preference or popularity among others are learned. The pipeline of our method is shown in Figure 1.1.

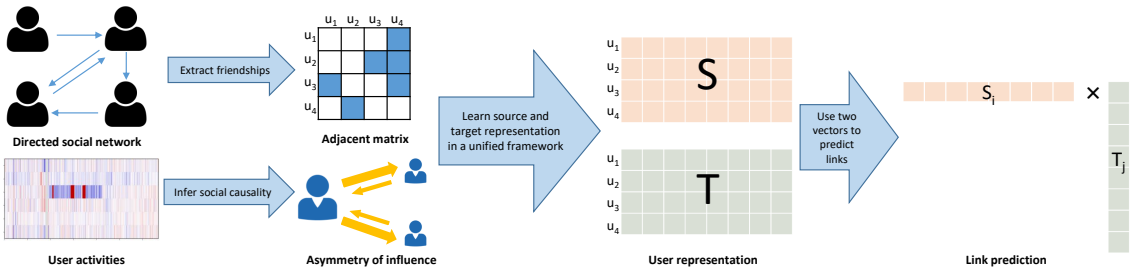

Figure 1.1: Pipeline of proposed framework

1.6 Summary of Main Contributions

The main contributions of this paper are:

- We propose a framework to incorporate social influence model into directed link prediction problem, where social influence can reflect rich set of user activity information.

- Our method gives each user two representations for a personalized social influence schema. It preserves the subtle differences of each user's interactions with different

[1]Source represents one user to follow others, characterizing the features as a follower; Target represents one user being followed by others, characterizing the features as a friend.

neighbors, which cannot be reflected by traditional method with a universal influence assignment.

- We propose to use an information-theoretic method to integrate heterogeneous social activities for inferring social influence in a general model without employing any domain-specific feature selection process.

- We conduct extensive experiments to verify that the proposed method can preserve rich information in directed social media better than the traditional methods, and therefore could be better used in directed link prediction.

1.7 Problem Statement

In this section, we introduce the notations and terminologies used in the paper and then formally define the link prediction problem in social networks. We use boldface uppercase letters (e.g. $\boldsymbol{X}$) to denote matrices and boldface lowercase letters to denote vectors (e.g. $\boldsymbol{x}$). We use $\boldsymbol{X}_i$ to denote the i^{th} row of the matrix, and $\boldsymbol{X}_{i,j}$ to signify the element in the i^{th} row and j^{th} column of $\boldsymbol{X}$. The transpose of $\boldsymbol{X}$ is represented as $\boldsymbol{X}^\top$. The ℓ_2-norm of a vector is represented by $||\cdot||_2$, and the Frobenius norm of a matrix is denoted as $||\cdot||_F$.

Let $\mathcal{G} = \{\mathcal{U}, \mathcal{E}\}$ be a directed network, where $\mathcal{U}$ indicates a set of N users $\{u_1, u_2, ..., u_N\}$ and $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{U}$ indicates the corresponding edge set. We denote a directed edge from u_i to u_j as $(i, j) \in \mathcal{E}$. Let $\mathcal{A}$ be a set of N sequences of timestamps. For each user u_i, the timestamp sequence $\mathcal{A}^{(i)} = \{t_{i,1}, t_{i,2}, ...\}$ records the occurrence time of all his/her online social activities. The time intervals of these activities could vary from seconds to months. Based on the terminologies defined above, we formally define the link prediction problem in social networks as follows:

Given a directed graph $\mathcal{G}$ associated with a set of edges $\mathcal{E}$ and a set of timestamp sequences $\mathcal{A}$ that records the occurrence of all users' social activities, we aim to predict

the probability of having a directed edge from any user u_i to any other user u_j, jointly based on the network topological structure in $\mathcal{G}$ and social activity information in $\mathcal{A}$.

2. RELATED WORK

For link prediction in social networks, people have been drawn attentions to it after the innovative work by Liben-Nowell and Kleinberg [5]. In general, most of the approaches are designed to calculate different kinds of proximity on social networks as prediction features [31, 1], where the learning framework includes supervised [32] and unsupervised [5]. In social networks, directed links are of great interest in many studies. Valverde-Rebaza and Lopes [33] proposed to combine community information with topology to predict links in a directed and asymmetric social network. Hopcroft and Tang [34] studied the reciprocal relationship prediction in directed social networks.

Methods with social influence model are well-studied in various domains, such as sociology and marketing literature [35, 36]. The social influence study often focuses on finding the most influential nodes, which has been applied to different applications, such as emotional contagion [11], and information diffusion by using IC (Independent Cascade) model [37].

3. METHODOLOGY: LINK PREDICTION WITH PERSONALIZED SOCIAL INFLUENCE (PSI)

3.1 Overview of PSI

To jointly model the topological structure and social activity information, we propose a link prediction framework named Personalized Social Influence (PSI).

The main idea is to learn two low-dimensional vector representations for each user u_i, i.e., Source representation $\boldsymbol{S}_i \in \mathbb{R}^{1\times d}$ and Target representation $\boldsymbol{T}_i \in \mathbb{R}^{1\times d}$, such that all the social influence among linked users is well preserved. The influence is directional and its strength depends on the social activity information in $\mathcal{A}$. Thus, we have two vector representations for each user, aiming to represent its roles in being affected and giving impacts respectively.

The proposed framework PSI could be separated into three major components as follows. First, it quantifies the strength of social influence based on the underlying patterns in the occurrence time of user activities. Second, it jointly embeds the directional network structure and the learned social activity information into two low-dimensional representations $\boldsymbol{S}$ and $\boldsymbol{T}$. Third, it accelerates the optimization via the Negative Sampling technique [15]. As a result, we could predict the probability of having a link from any user u_i to any other user u_j based on the inner product of two representations $\boldsymbol{S}_i\boldsymbol{T}_j$, as shown in Figure 1.1.

For the rest of this chapter, we will first introduce the intuition of our method that uses two personalized vector representations. Next we will introduce how to quantify the social influence from the heterogeneous social activity information. Then we will introduce how to jointly model social activity information and network structure information together. At last, we will introduce the acceleration of our model by using Negative Sampling.

3.2 Intuition Behind Using Two Personalized Vector Representations

In our method, we need to learn two low-dimension representations of each user for link prediction, so that we can multiply these two representations to get the quantification of proximity, as shown in Figure 1.1. Next we will introduce the intuition of making use of two vector representations.

In a partially observed social network, the positive instances in the adjacency matrix (i.e., the edge set $\mathcal{E}$) are usually extremely sparse [16], which renders a low rank structure. In light of this, the adjacency matrix can be approximated by two low rank matrices. We define these two matrices as Source representation matrix $\boldsymbol{S} \in \mathbb{R}^{n \times d}$ and Target representation matrix $\boldsymbol{T} \in \mathbb{R}^{n \times d}$. If the adjacency matrix is symmetric, it's usually accepted that $\boldsymbol{S} = \boldsymbol{T}$. But since we are approximating a directed graph, which doesn't have a symmetric adjacency matrix. That's why we make use of two distinct matrices. Our goal is to use the learned representations to approximate the proximity as accurate as possible.

On a user to user basis, when measuring the affinity of one user towards another user, we propose to use the product of the Source representation $\boldsymbol{S}$ and Target representation $\boldsymbol{T}$. We therefore define the rating of user u_i following user u_j:

$$\hat{r}(u_i, u_j) = \boldsymbol{S}_i \boldsymbol{T}_j \tag{3.1}$$

$r(\cdot, \cdot)$ means the rating of two users having a directed link, from the first user to the second user. We aim to distinguish the true friend of one user from a random user, i.e., the rating between two friends (positive instance) should be larger than the rating between two random users (negative instance). So that condition characterizes the following inequality:

$$\hat{r}(u_i, u_j) > \hat{r}(u_i, u_n) \qquad (i, j) \in \mathcal{E}, (i, n) \notin \mathcal{E}, \forall u_i \in \mathcal{U} \tag{3.2}$$

It means for any user as u_i, the rating with true friends u_j should be larger than the rating with negative users u_n. We use a probabilistic model to describe the intuition behind the link prediction using two personalized vectors of each user. The model is formulated as a Bayesian form to output the vectors, which are calculated in a pair-wise comparison manner. We can rewrite the optimization for the inequality in Eq. (3.2) as maximizing the posterior distribution:

$$P(\boldsymbol{S}, \boldsymbol{T}| >_u, \mathcal{G}) \propto P(>_u, \mathcal{G}|\boldsymbol{S}, \boldsymbol{T})P(\boldsymbol{S})P(\boldsymbol{T}) \tag{3.3}$$

where $>_u$ means the order of rating described in Eq. (3.2) as the intrinsic property in graph $\mathcal{G}$. Specifically, a user will prefer her true friend than a random user. Assume each user acts independently and each pair of users is compared independently, then the last term above can be written as:

$$\begin{aligned} &P(>_u, \mathcal{G}|\boldsymbol{S}, \boldsymbol{T})P(\boldsymbol{S})P(\boldsymbol{T}) \\ =&\Pi_{u_i}\Pi_{(i,j)\in\boldsymbol{E}}\Pi_{(i,n)\notin\boldsymbol{E}}P(>_{u_i}, \mathcal{G}|\boldsymbol{S}_i, \boldsymbol{T}_j, \boldsymbol{T}_n)P(\boldsymbol{S}_i)P(\boldsymbol{T}_j)P(\boldsymbol{T}_n). \end{aligned} \tag{3.4}$$

The probabilistic model is not intuitive to calculate due to that the greater-than sign $>_u$ doesn't have connection to two vector representations $\boldsymbol{S}$ and $\boldsymbol{T}$, so we adopt the concept of AUC (Area Under the Curve) to explain it. The greater-than sign is compared in a pair-wise manner, i.e., given two potential friends, the better model will predict the true friend over the random user. When we apply this pair-wise comparison to the scale of the whole network, we then can get the AUC value:

$$AUC = \frac{\sum\limits_{u_i\in\mathcal{U}}\sum\limits_{(i,j)\in\mathcal{E}}\sum\limits_{(i,n)\notin\mathcal{E}} I(\hat{r}(u_i, u_j) > \hat{r}(u_i, u_n))}{\sum\limits_{u_i\in\mathcal{U}}|P_{u_i}||N_{u_i}|}, \tag{3.5}$$

where $|P_{u_i}|$ and $|N_{u_i}|$ represents the number of positive instance and negative instances w.r.t user u_i. Specifically, the set $|P_{u_i}|$ contains the true friends of u_i and the set $|N_{u_i}|$ contains users that are not friends of u_i. $I()$ is the indicator function. The greater-than sign inside the indicator function represents the user preferences in the whole network, which is equal to Eq. (3.2). So the above optimization of probabilistic equation can be explained as optimizing the AUC curve.

Therefore, we build the connection between the greater-than sign in probabilistic model in Eq. (3.4) and two vector representations $\boldsymbol{S}$, $\boldsymbol{T}$. So, to achieve the probability distribution in Eq. (3.4) is equal to optimize the indicator function $I(\boldsymbol{S}_i\boldsymbol{T}_j > \boldsymbol{S}_i\boldsymbol{T}_n)$. Specifically, when the condition inside the indicator function $I(\cdot)$ holds true, it will give possibility as 1, and 0 otherwise. When we need to maximize the posterior probability in Eq. (3.4), we should apply the continuous probability function for calculating the derivatives. Some previous work such as that of Rendle et al. [17] suggests using sigmoid function to derive a probabilistic outcome,

$$\sigma(x) = 1/(1 + e^{-x}), \tag{3.6}$$

which is the ideal smooth version of 0/1 loss. So the indicator function in Eq. (3.5) can be rewritten as,

$$\Pi_{(i,j,n)}\sigma(\boldsymbol{S}_i\boldsymbol{T}_j - \boldsymbol{S}_i\boldsymbol{T}_n) \qquad (i,j) \in \mathcal{E}, (i,n) \notin \mathcal{E}, \forall u_i \in \mathcal{U} \tag{3.7}$$

After we ignore some constants, the objective can be written as,

$$\begin{aligned} \max_{\boldsymbol{S},\boldsymbol{T}} \quad \mathcal{J}_0 =& P(\boldsymbol{S}, \boldsymbol{T}| >_u, \mathcal{G}) \\ &\propto \prod_{(i,j)\in\mathcal{E},(i,n)\notin\mathcal{E},\forall u_i\in\mathcal{U}} \sigma(\boldsymbol{S}_i\boldsymbol{T}_j - \boldsymbol{S}_i\boldsymbol{T}_n). \end{aligned} \tag{3.8}$$

So we can see, the original probabilistic model will be transformed as optimizing through

a sigmoid function, which is essentially optimizing a AUC curve. Since the multiplication operator will suffer the problem of zero entries (i.e., if one entry is zero, the whole multiplication will be zero), we add a log in front of it. Therefore, the objective can be further rewritten as:

$$\max_{\boldsymbol{S},\boldsymbol{T}} \quad \mathcal{J}_0 = \sum_{(i,j,n)} \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j - \boldsymbol{S}_i \boldsymbol{T}_n). \tag{3.9}$$

So we explain the intuition behind using two personalized vector representations of each user. I.e. we can conduct a pair-wise comparison of users when they interact with different neighbors by taking different roles, which can be Source or Target in a directed network. But we haven't incorporated social activities of each user into the model. So next, we will introduce the quantification of social influence derived from social activities, and incorporate that information into the framework of using two personalized vector representations.

3.3 Social Influence Quantification

We now introduce how to model the social activity information. In this paper, we focus on the timestamp information, and the main reason is that other types of information of social activities could be heterogeneous and usually text-based, whose processing is computationally expensive. This is a trade-off between information of each user and the sheer quantity of the users available. I.e. if we want to incorporate more information from each user, which is usually the text-based information, we then can't incorporate a large quantity of users, since we need to design the Natural Language Processing (NLP) algorithms to text-based information to extract features for each individual user. The methods that use text-based information usually don't have large scale of online users. For example, Weng et al. [18] tried to find out the topical leaders in Twitter, and their dataset has 6748 users. Zhu et al. [19] were aiming to measure influence among users based on user generated content, and they consider 4013 users.

However, in this paper, we aim to incorporate as many online users as possible so

that we can understand the hidden reasons for the formation of links in social networks panoramically.

To model the timestamp information, an intuitive solution is to consider the timestamps as feature vectors and stack up every user's vector into a big feature matrix. However, this solution is not applicable to our problem since the feature matrix would be extremely sparse, due to that some users' may only have a few activity timestamps in their sequence $\mathcal{A}$, but some others may have a rich set of activity information. It implies that we can't treat every user's timestamps universally.

To quantify the social influence based on user activity information, we first personalize the timestamp sequence set $\mathcal{A}$ as a matrix $\boldsymbol{A}^{(i)}$ for each user u_i, so each user will have a unique matrix, which is later used in personalized learning.

3.3.1 Timestamp sequence modeling

For each user u_i, we have a timestamp sequence $\{t_{i,1}, t_{i,2}, ...\}$ that records the occurrence time of all his/her activities. Since the active periods of different users are quite diverse, we define a personalized time interval Δt^i for user u_i as follows.

$$\Delta t^i = (t_{max} - t_{min})/M, \tag{3.10}$$

where t_{min} and t_{max} denote the timestamps of the first and last activities of u_i, and M is a predefined maximum number of time intervals that is unique to each user, i.e. each user's maximum number of activities.

In order to quantify the personalized social influence one user receives, we will use his/her personalized time interval to separate all other users' timestamps sequences, so that we can apply some quantifications to calculate the personalized social influence specific to that user.

Next we will introduce the personalization of the activity matrix. We define the activity

frequency of u_i in the m^{th} interval as $\boldsymbol{A}_{i,m}^{(i)}$, given her own personalized time interval Δt^i. Note the activity matrix has been personalized based on the time interval Δt^i as $\boldsymbol{A}^{(i)}$, which will be used to calculate the personalized social influence that user u_i receives. Similarly $\boldsymbol{A}_{i,m}^{(j)}$ denotes u_i's activities at m^{th} time interval, which is derived from u_j's personalized time interval. Then the sequence set $\mathcal{A}$ is personalized as a set of matrices $\boldsymbol{A}^{(i)} \in \mathbb{R}^{N \times M}$ for each user. It should be noted that, for different users, their activity time intervals are different, varying from seconds to months. Thus, we have N different personalized social activity matrices $\{\boldsymbol{A}^{(i)}\}$, for $i = 1, 2, ..., N$.

3.3.1.1 Metric of social influence

We now introduce a pair-wise manner to calculate personalized social influence. We focus on a pair of linked users, with a directional edge (i, j) denoting u_i following u_j, where the influence is actually from u_j to u_i. We quantify this social influence by measuring the reduction of uncertainty of $\boldsymbol{A}_i^{(i)}$ given the knowledge of $\boldsymbol{A}_j^{(i)}$. Note we use the personalized time interval of the user being influenced, which is u_i. And $\boldsymbol{A}_i^{(i)}$, $\boldsymbol{A}_j^{(i)}$ are the time-series vectors in the personalization of u_i. We infer the influence by comparing two scenarios.

(1) We aim to calculate the dependency of u_i on herself. By considering vector $\boldsymbol{A}_i^{(i)}$ as a Markov chain with M variables, we could infer the activity frequency $\boldsymbol{A}_{i,m+1}^{(i)}$ based on the historical record $\boldsymbol{A}_{i,m}^{(i)}$. We denote the probability as,

$$\boldsymbol{p}_1 = \left\{ \mathrm{P}(\boldsymbol{A}_{i,m+1}^{(i)} \neq 0 | \boldsymbol{A}_{i,m}^{(i)}) \right\}, \text{ for } m = 1, ..., M-1. \tag{3.11}$$

So we could get a probability distribution $\boldsymbol{p}_1$ with $M-1$ probability values. The intuition of $\boldsymbol{p}_1$ is the probability that we can use u_i's own history to predict her future activities.

(2) We assume that user u_j has influence on user u_i. This influence could often be reflected as the dependency from activity frequency $\boldsymbol{A}_i^{(i)}$ to $\boldsymbol{A}_j^{(i)}$ [20]. Mathematically, the

probability of having influence between the two is now defined as follows,

$$\boldsymbol{p}_2 = \left\{ \mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}, \boldsymbol{A}^{(i)}_{j,m}) \right\}, \text{ for } m = 1, ..., M-1. \tag{3.12}$$

Thus, we get another probability distribution $\boldsymbol{p}_2$, which measures how well we can predict u_i's activity based on her own history and a potential influencer u_j.

It should be noted that we only consider the influence among adjacent time intervals, e.g. m and $m+1$, since the influence is often stronger than the non-adjacent one. Motivated by the study of influence flow [21], we employ the entropy reduction of $\boldsymbol{p}_1$ from $\boldsymbol{p}_2$ as a metric of social influence from u_j to u_i, i.e.,

$$\begin{aligned} I_{j\to i} \triangleq & H(\boldsymbol{p}_1) - H(\boldsymbol{p}_2) \\ = & -\sum_{m=1}^{M-1} \mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}) \log \mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}) \\ & + \sum_{m=1}^{M-1} \mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}, \boldsymbol{A}^{(i)}_{j,m}) \log \mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}, \boldsymbol{A}^{(i)}_{j,m}), \end{aligned} \tag{3.13}$$

where $H(\boldsymbol{p}_1)$ and $H(\boldsymbol{p}_2)$ denote the entropy of distributions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. The key idea of metric $I_{j\to i}$ is that, when u_j has influence on u_i, then $\boldsymbol{A}^{(i)}_j$ could reduce the uncertainty of predicting $\boldsymbol{A}^{(i)}_i$. Therefore, we have mathematically calculated social influence in a personalized schema.

3.3.1.2 Computation of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$

It should be noted that $\mathbf{p}_1$ and $\mathbf{p}_2$ are two sets of conditional probabilities. First, we employ a logistic regression model to calculate the conditional probability of u_i being active given her own history,

$$\mathrm{P}(\boldsymbol{A}^{(i)}_{i,m+1} \neq 0 | \boldsymbol{A}^{(i)}_{i,m}) = \frac{1}{1 + e^{-\alpha_0 - \alpha_1 f(\boldsymbol{A}^{(i)}_{i,m})}}, \tag{3.14}$$

where α_0 and α_1 are coefficients learned from the entire record. Specifically, we employ the sequence $\{\boldsymbol{A}_{i,1}^{(i)}, \boldsymbol{A}_{i,2}^{(i)}, ..., \boldsymbol{A}_{i,M-1}^{(i)}\}$ as $M-1$ training inputs, and the binary sequence $\{\text{sgn}(\boldsymbol{A}_{i,2}^{(i)}), \text{sgn}(\boldsymbol{A}_{i,3}^{(i)}), ..., \text{sgn}(\boldsymbol{A}_{i,M}^{(i)})\}$ as corresponding labels, where function $\text{sgn}(\cdot)$ is the sign function. We also employ a discount function $f(x)$ for each activity frequency, which is defined as follows.

$$f(x) = \begin{cases} x, & \text{if } x \leq 2, \\ 1 + \lceil \log(1+x) \rceil, & \text{o/w.} \end{cases} \tag{3.15}$$

The basic idea behind $f(x)$ is that as more activities occurred during only one time interval Δt^i, the number of activities actually seen by people would not increase linearly. For instance, in Twitter, the activity frequency could be the number of tweets that a user u_j has posted during one time interval Δt^i. If u_j posts a large number of tweets in a short period, his/her follower u_i may not check all of them [22]. Thus we employ a discount function $f(x)$ to estimate the actual number of activities that could be perceived by u_i.

Similarly, we could calculate $\boldsymbol{p}_2$, with pre-trained coefficients α_0, α_1, and α_2, i.e.,

$$\text{P}(\boldsymbol{A}_{i,m+1}^{(i)} \neq 0 | \boldsymbol{A}_{i,m}^{(i)}, \boldsymbol{A}_{j,m}^{(i)}) = \frac{1}{1+e^{-\alpha_0 - \alpha_1 f(\boldsymbol{A}_{i,m}^{(i)}) - \alpha_2 f(\boldsymbol{A}_{j,m}^{(i)})}}. \tag{3.16}$$

By substituting $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ into Eq. (3.13), we can calculate the reduction of uncertainty of u_i's activities given the knowledge of u_j's activities, which is defined as the personalized social influence in this paper.

3.4 Jointly Modeling Social Activity and Network Structure

Since each user u_i could both be influenced by others and give influence to others, we employ two vector representations to represent u_i, i.e., Source representation $\boldsymbol{S}_i$ and Target representation $\boldsymbol{T}_i$. In such way, we could predict the probability of having a link

from any user u_i to any other user u_j based on the inner product of two representations, $\boldsymbol{S}_i\boldsymbol{T}_j$. The main goal is to make sure $\boldsymbol{S}_i\boldsymbol{T}_j > \boldsymbol{S}_i\boldsymbol{T}_n$ for any pair of existing edge $(i,j) \in \mathcal{E}$ and non-existent edge $(i,n) \notin \mathcal{E}$.

A traditional method is to focus on existing edges and make the estimated probability of u_i following u_j approach the probability determined by the edge weights [23]. However, it could not be directly applied to our problem, since it could not take advantage of user activity information. To jointly embed the social activity information and network structure, we propose to estimate the linking probabilities based on the learned social influence.

We now define an empirical probability of having a directed edge from user u_i to u_j as follows. The main idea is to calculate the amount of influence solely from u_j towards u_i, comparing with all other potentially influential users.

$$\hat{p}(u_j|u_i) = I_{j\to i}/d_i^{out}, \tag{3.17}$$

where $d_i^{out} = \sum_{(i,l)\in\hat{\mathcal{E}}} I_{l\to i}$ is the out-degree of user u_i[1], which in social influence scenario, is the set of users who influence u_i. Note we define a new set of edges $\hat{\mathcal{E}}$ that only includes node pairs with significant influence, i.e., we only consider node pairs with social influence greater than a threshold,

$$I_{j\to i} > \frac{\log_2(\sum_{m=1}^{M} \boldsymbol{A}_{i,m})}{2\sum_{m=1}^{M} \boldsymbol{A}_{i,m}}. \tag{3.18}$$

The threshold is motivated by Minimum Description Length penalty (MDL) [24]. The main reason for applying a threshold here is that more active users will be more likely to overfit the model, but less active users may not be learned properly.

We employ the softmax function to calculate the probability of having a directed edge

[1]Out-degree of u_i represents the set of users that u_i is following.

from user u_i to u_j through their representations,

$$p(u_j|u_i) = \frac{e^{\boldsymbol{S}_i \boldsymbol{T}_j^\top}}{\sum_{l=1}^{N} e^{\boldsymbol{S}_i \boldsymbol{T}_l^\top}}. \tag{3.19}$$

The key idea is to determine the linking probability based on the contribution of user u_i to u_j comparing with all other users.

Therefore, by minimizing the Kullback–Leibler divergence of $p(u_j|u_i)$ and its empirical counterpart $\hat{p}(u_j|u_i)$, we can get the objective function:

$$\begin{aligned} \min \quad \mathcal{J}_1 &= \sum_{(i,j)\in\hat{\mathcal{E}}} d_i^{out} D_{\mathrm{KL}}(\hat{p}(u_j|u_i)||p(u_j|u_i)) \\ &= -\sum_{(i,j)\in\hat{\mathcal{E}}} I_{j\to i} \log p(u_j|u_i) + \frac{I_{j\to i}}{d_i^{out}} \log \frac{I_{j\to i}}{d_i^{out}}, \end{aligned} \tag{3.20}$$

where d_i^{out} represents the prestige of user u_i in the network, which is defined before. After ignoring the constant, the objective can be simply rewritten as:

$$\max \quad \hat{\mathcal{J}}_1 = \sum_{(i,j)\in\hat{\mathcal{E}}} I_{j\to i} \log p(u_j|u_i). \tag{3.21}$$

Since we want to minimize the divergence, it is equivalent to maximizing the objective by omitting the negative sign.

3.5 Acceleration via Negative Sampling

We can see optimizing $\hat{\mathcal{J}}_1$ is computationally expensive since by calculating Eq. (3.21) we need to calculate one softmax as Eq. (3.19), i.e., every pair of users needs to compare with all the users. So before we optimize $\hat{\mathcal{J}}_1$, we would like to rewrite Eq. (3.19) as

follows.

$$p(u_j|u_i) = \frac{1}{1 + \sum_{l=1, l\neq j}^{N} e^{-(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_l^\top)}}. \tag{3.22}$$

It follows the format of Eq. (3.19), just by dividing both numerator and denominator by $e^{\boldsymbol{S}_i \boldsymbol{T}_j^\top}$. Note the form of sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, we can define a new conditional probability in the similar form of Eq. (3.22):

$$p(u_j > u_n|u_i) = \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top) = \frac{1}{1 + e^{-(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)}}. \tag{3.23}$$

The above conditional probability can be interpreted as instead of directly optimizing $\hat{\mathcal{J}}_1$ over all users, we update Eq. (3.23) with respect to a small set of noise samples in $\mathcal{U}\backslash j$, where an individual sample is denoted as u_n [25]. It can be easily verified that:

$$p(u_j|u_i) > \prod_{u_n \in \mathcal{U}\backslash j} p(u_j > u_n|u_i). \tag{3.24}$$

Therefore, instead of optimizing $\hat{\mathcal{J}}_1$, we can optimize a tight lower bound of $p(u_j|u_i)$ in $\hat{\mathcal{J}}_1$. So if we combine Eqs. (3.23) and (3.24) and put them back to objective $\hat{\mathcal{J}}_1$ in Eq. (3.21), we can get a new objective function:

$$\max \quad \mathcal{J}_2 = \sum_{(i,j)\in\hat{\mathcal{E}}} I_{j\to i} \sum_{u_n \in \mathcal{U}\backslash j} \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top). \tag{3.25}$$

It should be noted that the $I_{j\to i}$ needs to satisfy Eq. (3.18). To accelerate the learning process, here we adopted Negative Sampling in [15]. All the negative samples will be

drawn from a noise distribution. So the probability part can be further rewritten as:

$$\begin{aligned}&\sum_{u_n \in \mathcal{U} \setminus j} \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top) \\ &\propto \sum_{n}^{K} E_{u_n \sim P_n(u)} \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top).\end{aligned} \tag{3.26}$$

where K is the number of negative instances, sampled from noise distribution of $P_n(u) \propto d_u^{3/4}$, and d_u is the out-degree of user u. Thus, there is no need to go through all users to get the conditional probability, but just fewer noise samples.

We can then write the final objective function in a unified from, incorporating network structure and social influence with acceleration learning schema:

$$\begin{aligned}\max \ \ \mathcal{J} = &\sum_{(i,j) \in \hat{\mathcal{E}}} I_{j \to i} \sum_{n}^{K} E_{u_n \sim P_n(u)} \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top) \\ &- \frac{\beta_1}{2} ||\boldsymbol{S}||_F^2 - \frac{\beta_2}{2} ||\boldsymbol{T}||_F^2,\end{aligned} \tag{3.27}$$

The last two terms $||\boldsymbol{S}||_F^2$ and $||\boldsymbol{T}||_F^2$ are employed to avoid overfitting. Since we aim to maximize the objective, the overfitting term will take negative sign. β_1 and β_2 are the regularization coefficients.

3.6 Computation of $\boldsymbol{S}$ and $\boldsymbol{T}$

Next we will derive the update rules of each model parameter $\Theta = \boldsymbol{S}$ or $\boldsymbol{T}$. In each iteration, we can update model parameter according to asynchronous stochastic gradient descent (ASGD), which is a fast optimization algorithm in many machine learning applications. For each model parameter Θ, we derive the gradient from Eq. (3.27) as follows.

$$\frac{\partial \mathcal{J}}{\partial \Theta} = \sum_{(i,j) \in \hat{\mathcal{E}}} I_{j \to i} \sum_{n}^{K} E_{u_n \sim P_n(u)} \frac{\partial \mathcal{L}(u_i, u_j, u_n)}{\partial \Theta} - \beta_\Theta \Theta, \tag{3.28}$$

where we define $\mathcal{L}(u_i, u_j, u_n) = \log \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)$. Therefore, we only need to calculate the derivative of $\mathcal{L}(u_i, u_j, u_n)$ w.r.t. each $\Theta = \boldsymbol{S}$ or $\boldsymbol{T}$ since other calculation is the same for each model parameter. So for user Source representation $\boldsymbol{S}$, we have,

$$\frac{\partial \mathcal{L}(u_i, u_j, u_n)}{\partial \boldsymbol{S}_i} = \epsilon \frac{\partial \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)}{\partial \boldsymbol{S}_i}, \tag{3.29}$$

where ϵ is defined as $\epsilon = 1/\sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)$. As for user Target representation $\boldsymbol{T}$, we have,

$$\begin{aligned} \frac{\partial \mathcal{L}(u_i, u_j, u_n)}{\partial \boldsymbol{T}_j} &= \epsilon \frac{\partial \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)}{\partial \boldsymbol{T}_j}, \\ \frac{\partial \mathcal{L}(u_i, u_j, u_n)}{\partial \boldsymbol{T}_n} &= -\epsilon \frac{\partial \sigma(\boldsymbol{S}_i \boldsymbol{T}_j^\top - \boldsymbol{S}_i \boldsymbol{T}_n^\top)}{\partial \boldsymbol{T}_n}. \end{aligned} \tag{3.30}$$

For simplicity, we omit the writing of regularization term $\beta_\Theta \Theta$ of each model parameter.

The whole process to learn the user representation is shown in Algorithm 1.

3.7 Complexity Analysis

We only need $O(|U|d + |E|)$ space overheads since we adopt the per-observation stochastic gradient updates on the fly, where $|U|$ is the user set, d is the representation dimension, E is the edge set. As for time complexity, since we use negative sampling, the posterior possibility can be largely reduced from $O(|E|)$ to $O(1)$, because we repeatedly draw negative samplings from the same noise distribution. So for the negative samples of each edge, we need $O(d \times (|K|+1))$ time, where K is the negative samplings. Usually the number of iterations needed for optimization is proportional to the number of edges. So the final time complexity for our model would be $O(d|K||E||L|)$, where L is the average friends of one user.

Algorithm 1: Representation Learning Algorithm

Input : edge set $\mathcal{E}$, user set $\mathcal{U}$, user activities $\mathcal{A}$
Output: user source representation $\boldsymbol{S}$ and user target representation $\boldsymbol{T}$
Initialize $\boldsymbol{S}$, $\boldsymbol{T}$
while *not converge* **do**
  for each $u_i \in \mathcal{U}$ **do**
    Derive personalized time interval of u_i as Δt^i using Eq. (3.10)
    Separate all other user's time sequence $\mathcal{A}$ to personalized matrix $\boldsymbol{A}^{(i)}$ using Δt^i
    for each u_j in $(i, j) \in [\mathcal{E}]$ **do**
      Calculate two probability distributions of Eq. (3.11) and Eq. (3.12)
      Based on two distributions, calculate the difference of two entropy using Eq.(3.13)
    end
    Draw $\boldsymbol{K}$ negative samples from distribution $P_n(u)$
    for each $n \in \boldsymbol{K}$ **do**
      Update user source representation $\boldsymbol{S}_i$ and user target representations $\boldsymbol{T}_j$, $\boldsymbol{T}_n$ according to Eq.(3.28), Eq. (3.29) and Eq. (3.30)
    end
  end
end
return $\boldsymbol{S}$ and $\boldsymbol{T}$

4. EXPERIMENT

In this section, we empirically evaluate our method by comparing it with the state-of-the-art methods. We aim to answer two questions: (1) What is the impact of learning social influence on performing link prediction? (2) How effective is our method in modeling social activity information, especially when it is sparse?

4.1 Datasets

We use two publicly available datasets in our experiment: URL Twitter dataset [26] and Higgs Twitter dataset [27]. The URL dataset was collected by tracking the tweets with diffrent URL links on Twitter. The users that posted all the predefied set of URLs are crawled, with their following relationship with each other. The second dataset is the Higgs Twitter dataset. It was built after monitoring the spreading processes on Twitter before, during and after the announcement of the discovery of a new particle with the features of the elusive Higgs boson. It collects all the tweets discussing this discovery, containing each users retweeting, replying and mentioning with others. The users IDs have been anonymized. The details of these two datasets are shown in Table 4.1.

Dataset	URL dataset	Higgs dataset
# of users	736,930	456,626
# of user activities	2,859,764	563,069
# of directed links	36,743,448	14,855,842

Table 4.1: Statistics of Experimental Dataset

4.2 Baseline Methods

We compare our algorithm in the directed link prediction problem with the state-of-the-art methods, which could be separated into three categories, specifically as methods with only social activities, methods with only network structure, and methods with both sources of information. First, we want to investigate the effectiveness of our method in learning the network structure, so we compare it with methods that learn from social activities, i.e. R&M. Second, we want to investigate the impact of considering social activity information for link prediction problem. We compare our method with baselines that only consider network structure, i.e. CN, BPR-MF, and ELLR. Third, we want to study the effectiveness of the proposed method in jointly learning social activities and network structure, so we compare our method with TI and SWR. The details of baselines are shown as follows.

- R&M [28]: The directed links are inferred by the counts of retweet and mention of each pair of users.
- CN [5]: The Common Neighbor method is widely adopted for link prediction problem, due to its simplicity for implementation.
- BPR-MF [17]: It is the Bayesian Personalized Ranking in matrix factorization framework for predicting the links between users.
- ELLR [2]: It uses a generalized AUC for an Efficient Latent Link Recommendation.
- TI [29]: The method exploits Temporal Influence for link prediction by using matrix factorization. The temporal information is based on time delay of each pair of users, where smaller delay means higher influence.
- SRW [6]: The Supervised Random Walks method learns the edge weights to let the random walker more likely to traverse nodes that have edges with current nodes. We

use concatenation of two users' activities record as an edge vector between them.

4.3 Experimental Setup and Evaluation Metrics

We conduct our experiment from two aspects which are well established in link prediction problems to test the algorithm performance, i.e. *pair-wise* accuracy and *list-wise* accuracy [30].

First, to test pair-wise accuracy, each test instance is a tuple with three users, i.e. a user, her true friend (positive instance) and a random user (negative instance). We aim to measure whether the algorithm can distinguish a positive link from the negative one, i.e. pair-wise accuracy. And we average the accuracy of all the test instances to have the final pair-wise accuracy. The metric we adopted is AUC (Area Under the Curve),

$$AUC = \frac{\sum_{u_i \in \mathcal{U}} \sum_{(i,j) \in \mathcal{E}} \sum_{(i,n) \notin \mathcal{E}} I(\hat{r}(u_i, u_j) > \hat{r}(u_i, u_n))}{\sum_{u_i \in \mathcal{U}} |P_{u_i}||N_{u_i}|}, \tag{4.1}$$

where $|P_{u_i}|$ and $|N_{u_i}|$ represents the number of positive instance and negative instances w.r.t user u_i. $I()$ is the indicator function. AUC is suitable for test the data which is highly imbalanced, since in our dataset the negative instances are more ubiquitous than positive instances. In terms of separating training and testing set, we randomly select different subset of tuples from dataset (i.e., 10%, 20%, 40%, 60%) to train the model. For each fraction of training set, all the remaining instances will be used as test set.

Second, the list-wise accuracy measures the portion of true friends in a ranked list of recommended friends returned by the algorithms, where better algorithm intuitively will give true friends higher rank in the list. So we adopt Precision@k to measure the accuracy in the ranked list of users in different positions,

$$\text{Precision@}k = \frac{\#\text{ of positive instances in the top } k}{\#\text{ of positive and negative instances in the top } k}, \tag{4.2}$$

and MAP (Mean Average Precision) which is averaged by all users in different positions from Precision@k for each dataset.

$$MAP = \frac{\sum_{u_i \in \mathcal{U}} \frac{\sum_{k=1}^{m} \text{Precision@}k}{m}}{|\mathcal{U}|}, \tag{4.3}$$

where k is the positions at the estimated rank list, $\mathcal{U}$ is the set of all users. We evaluate the precision at the first 10 positions of the ranked list. In terms of separating training and testing set, all the directed links are randomly divided into two groups, where 60% is for training the model, and 40% is for testing the list-wise accuracy.

4.4 Hyper-Parameter Discussion

(1) Learning rate and regularization parameters. We conduct a grid search over candidate set of learning rate and regularization parameters. So we set the learning rate as 0.01 and regularization parameters of representations as 0.025.

(2) Likelihood function coefficients. Numerically, the coefficients $\alpha = \{\alpha_0, \alpha_1, \alpha_2\}$ in Eq. (3.16) can take positive or negative values. Intuitively, a positive coefficient α_2, for instance, corresponds to user j boosting user i's number of activities in next time point $m + 1$. However, a negative coefficient was more likely because of over-fitting than the situation that user j will suppress user i's activities. Therefore, if any coefficient happens to be negative in likelihood function, it will be rejected.

4.5 Experimental Results

We evaluate the results based on the questions that we asked before, by comparing the proposed method PSI with the baseline methods. Next we will discuss the experimental results in detail.

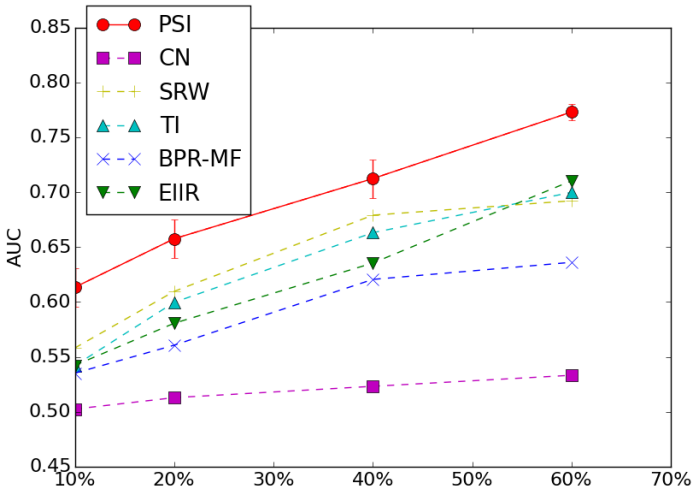

Figure 4.1: AUC on URL dataset

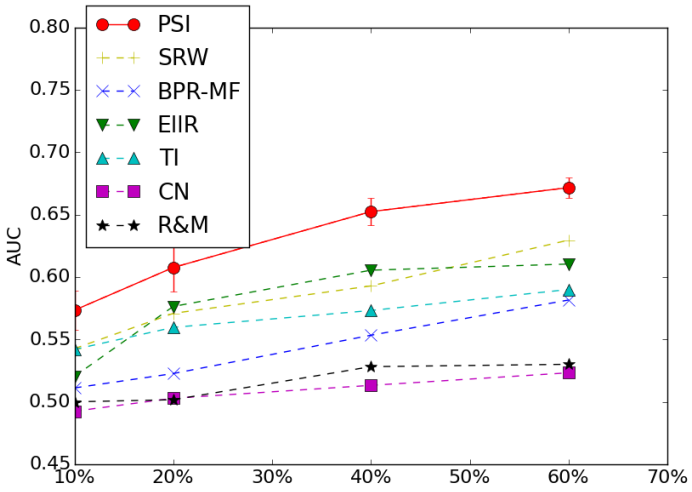

Figure 4.2: AUC on Higgs dataset

4.5.1 Effectiveness of jointly learning

We now answer the first question, i.e. how effective can our method jointly learn social activity and network structure information compared to other methods. We investigate this

question by looking into the two accuracy metrics.

A. Pair-wise accuracy. The results of pair-wise accuracy in terms of AUC metric are shown in Figure 4.1 and Figure 4.2. Note that URL dataset does not have information of retweet and mention, so we only conduct R&M on Higgs dataset with AUC metric. We have several observations as follows.

(1) Methods that only consider one source of information perform the worst in both datasets. For example, CN was out performed by others in a large margin, due to the fact that it only the neighbors of each user.

(2) In general, the methods which consider both user activity and network structure information outperformed the methods that only consider one of those. For instance, SWR achieves 18.7% and 20.3% gain over R&M and CN respectively with 60% training set in Higgs dataset. However, our method further achieves 6.68% gain over SWR, indicating that the PSI can better integrate social activities with network structure information.

(3) The proposed method PSI has better performance with smaller training set. For example, in URL dataset, with only 40% of training set our method has higher accuracy than most of the baselines with 60% training set. In Higgs dataset, our method even only requires 20% of training to outperform most of the baselines with 60% training set.

(4) Methods generally have better performance in URL dataset than in Higgs dataset. We infer that is due to the sparsity of Higgs dataset in terms both following relationships and user activities.

(5) While all other baseline methods can't perform stably across datasets, the proposed method PSI can consistently outperform other baseline methods in both datasets, which indicates its robustness in different real-world scenarios.

B. List-wise accuracy. We now investigate the results of list-wise accuracy by using the Precision@k which is shown Figure 4.3 and Figure 4.4. And MAP is shown in Figure 4.5 and Figure 4.6. We draw several observations as follows.

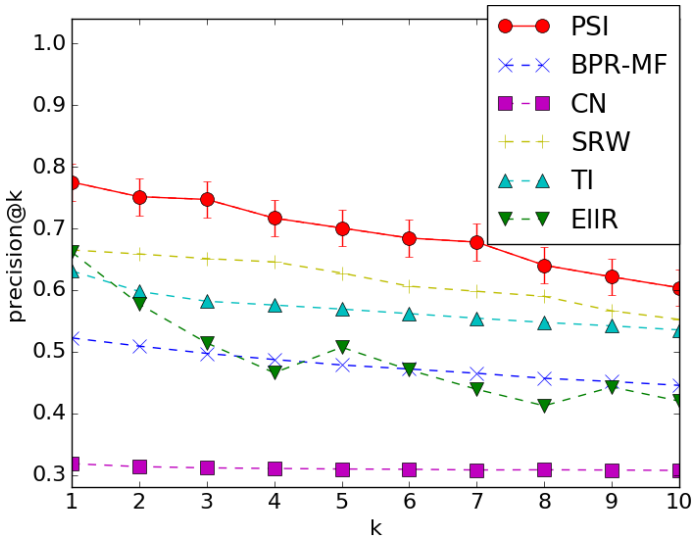

Figure 4.3: Precision@k on URL dataset

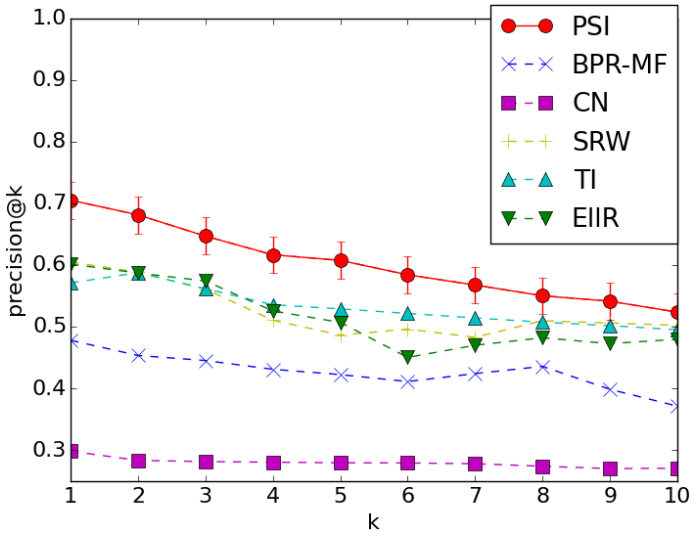

Figure 4.4: Precision@k on Higgs dataset

(1) Methods that combine both network structure and user activities generally outperform the methods that consider only one of those. But the improvement is more salient in

URL dataset than in Higgs dataset, again partially due to the sparsity of data.

(2) Our method performs relatively stable in different positions in Precision@k metric in Figure 4.3 and Figure 4.4, while some of the baselines did not, such as ELLR in URL dataset in position 4 to 9 as shown in Figure 4.3, and BPR-MF in Higgs dataset in position 6 to 9 as shown in Figure 4.4. We can see these two methods have drastic changes in those positions. We infer that without using social activities as the extra information to regularize the model, those methods with only network structure information will be largely affected by the noisy instances in recommending a list of users.

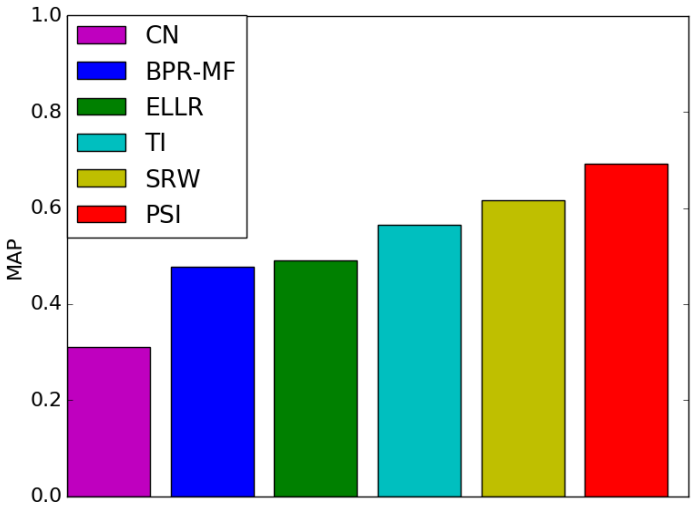

Figure 4.5: MAP on URL dataset

(3) The social activity information affects the models in an evident way. As shown in Figure 4.5 and Figure 4.6. In URL dataset, ELLR performs worse than TI and SRW with respect to MAP metric. However, in Higgs dataset where the social activities are relatively sparse, EllR even performs slightly better than these two methods, indicating these two methods cannot properly learn from sparse data. But our method consistently

outperforms those methods, meaning that it can better learn useful information from sparse social activities.

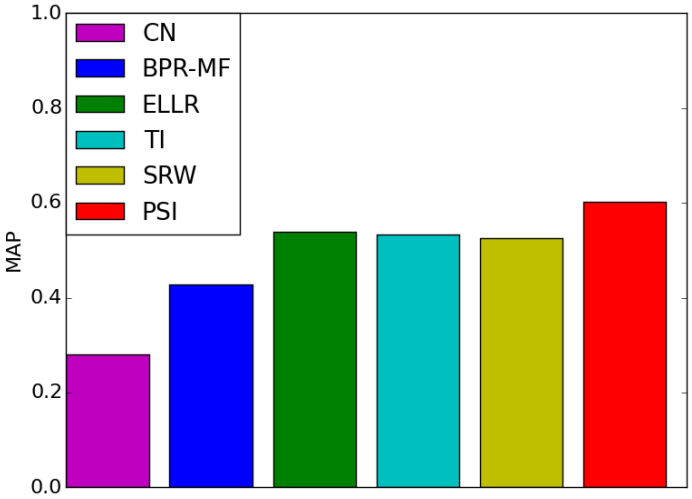

Figure 4.6: MAP on Higgs dataset

In a nutshell, our method PSI consistently achieves better performance in link prediction by jointly modeling social activity and network structure, which answers first question.

4.5.2 Impact of sparsity of social activities

We now answer the second question, i.e., how effective of our method in modeling social activity information, especially when it is sparse. We have two baseline methods, TI and SRW, which consider both network structure and user activity information. So we will compare our method with these two methods here.

We first sequentially (time sequence from oldest to newest) sample different portions of each user's activities, e.g., if we sample 10% of user activities, it means only the first 10% of social activities are included. We use MAP as the evaluation metric. The results

	10% (gain)	25% (gain)	50% (gain)	100% (gain)
TI	0.4813 (N.A.)	0.5187 (N.A.)	0.5323 (N.A.)	0.5647 (N.A.)
SRW	0.4998 (+3.87%)	0.5453 (+5.12%)	0.5738 (+7.79%)	0.6165 (+9.15%)
PSI	**0.543 (+12.91%)**	**0.596 (+14.97%)**	**0.633 (+18.94%)**	**0.691 (+22.51%)**

Table 4.2: Model Sensitivity on User Activity Size of URL dataset

	10% (gain)	25% (gain)	50% (gain)	100% (gain)
TI	0.3925 (N.A.)	0.4364 (N.A.)	0.4854 (N.A.)	0.5327 (N.A.)
SRW	0.4303 (+9.63%)	0.4455 (+2.08%)	0.4922 (+1.40%)	0.5250 (-1.44%)
PSI	**0.453 (+15.61%)**	**0.505 (+15.78%)**	**0.532 (+9.64%)**	**0.602 (+13.12%)**

Table 4.3: Model Sensitivity on User Activity Size of Higgs dataset

are shown in Table 4.2 and 4.3. Each column represents how much user activities are used, and 'gain' means the percentage improvement of the methods as compared to TI. The result shows the robustness of our method with sparse user activities. In URL dataset, compared with TI and SRW, when only 10% of user activities are considered, our method already outperforms TI and SRW by 12.91% and 8.80%, respectively. When we include all social activity information, our method can achieve 22.51% and 12.23% gain at maximum over TI and SWR. In Higgs dataset, we can observe the similar outcome, for only 10% activities considered, our method outperforms TI and SRW by 15.61% and 5.46%, and the maximum gain is achieved in 25% training set.

In summary, our method can properly learn useful information from social activities better than baselines methods, and thus performs better in link prediction, even with sparse activities data, which answers the second question before.

5. CONCLUSION

In this paper, we study link prediction in social networks by considering the social influence model. While link prediction has been intensively studied with pure network structures, we prove that by incorporating social influence measure into network topological structure, our method can perform better in link prediction. The quantification of social influence is learned from social activities through information-theoretic method, and the personalized social influence is further preserved in the user Source representation and Target representation, which can individually represent the users' personalized characteristics. In link prediction task, our method outperforms the state-of-the-art methods in directed link prediction, indicating the effectiveness of PSI in jointly learning social activity information and network topological structure in a unified framework.

REFERENCES

[1] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 243–252, ACM, 2010.

[2] D. Song, D. A. Meyer, and D. Tao, "Efficient latent link recommendation in signed networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1105–1114, ACM, 2015.

[3] N. Barbieri, F. Bonchi, and G. Manco, "Who to follow and why: link prediction with explanations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1266–1275, ACM, 2014.

[4] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 44–54, ACM, 2006.

[5] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *journal of the Association for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[6] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 635–644, ACM, 2011.

[7] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*, vol. 8. Cambridge university press, 1994.

[8] O. Tsur and A. Rappoport, "What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 643–652, ACM, 2012.

[9] P. V. Marsden and N. E. Friedkin, "Network studies of social influence," *Sociological Methods & Research*, vol. 22, no. 1, pp. 127–151, 1993.

[10] E. Bakshy, B. Karrer, and L. A. Adamic, "Social influence and the diffusion of user-created content," in *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 325–334, ACM, 2009.

[11] Y. Yang, J. Jia, B. Wu, and J. Tang, "Social role-aware emotion contagion in image social networks," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 807–816, ACM, 2009.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web.," 1999.

[14] R. S. Burt, *Structural holes: The social structure of competition*. Harvard university press, 2009.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[16] D. Song and D. A. Meyer, "Recommending positive links in signed social networks by optimizing a generalized auc.," in *AAAI*, pp. 290–296, 2015.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 452–461, AUAI Press, 2009.

[18] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 261–270, ACM, 2010.

[19] Z. Zhu, J. Su, and L. Kong, "Measuring influence in online social network based on the user-content bipartite graph," *Computers in Human Behavior*, vol. 52, pp. 184–189, 2015.

[20] G. Ver Steeg and A. Galstyan, "Information transfer in social media," in *Proceedings of the 21st international conference on World Wide Web*, pp. 509–518, ACM, 2012.

[21] G. Ver Steeg and A. Galstyan, "Information-theoretic measures of influence based on content dynamics," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 3–12, ACM, 2013.

[22] C. J. Quinn, N. Kiyavash, and T. P. Coleman, "Directed information graphs," *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6887–6909, 2015.

[23] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, International World Wide Web Conferences Steering Committee, 2015.

[24] P. D. Grünwald, *The minimum description length principle*. MIT press, 2007.

[25] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han, "Large-scale embedding learning in heterogeneous event data," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 907–912, IEEE, 2016.

[26] N. O. Hodas and K. Lerman, "The simple rules of social contagion," *Scientific reports*, vol. 4, 2014.

[27] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." `http://snap.stanford.edu/data`, June 2014.

[28] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy.," *ICWSM*, vol. 10, no. 10-17, p. 30, 2010.

[29] R. Pálovics, A. A. Benczúr, L. Kocsis, T. Kiss, and E. Frigó, "Exploiting temporal influence in online recommendation," in *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 273–280, ACM, 2014.

[30] W. Zhang, J. Wang, and W. Feng, "Combining latent factor model with location features for event-based group recommendation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 910–918, ACM, 2013.

[31] H. Chen, X. Li, and Z. Huang, "Link prediction approach to collaborative filtering," in *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on*, pp. 141–142, IEEE, 2005.

[32] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[33] J. Valverde-Rebaza and A. de Andrade Lopes, "Exploiting behaviors of communities of twitter users for link prediction," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1063–1074, 2013.

[34] J. Hopcroft, T. Lou, and J. Tang, "Who will follow you back?: reciprocal relationship prediction," in *Proceedings of the 20th ACM international conference on Information*

and knowledge management, pp. 1137–1146, ACM, 2011.

[35] M. Granovetter, "Threshold models of collective behavior," *American journal of sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[36] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing letters*, vol. 12, no. 3, pp. 211–223, 2001.

[37] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, ACM, 2003.