Structural Link Analysis and Prediction in Microblogs

Dawei Yin Liangjie Hong Brian D. Davison
Department of Computer Science & Engineering, Lehigh University
Bethlehem, PA 18015 USA
{day207, lih307, davison}@cse.lehigh.edu

ABSTRACT

With hundreds of millions of participants, social media services have become commonplace. Unlike a traditional social network service, a microblogging network like Twitter is a hybrid network, combining aspects of both social networks and information networks. Understanding the structure of such hybrid networks and predicting new links are important for many tasks such as friend recommendation, community detection, and modeling network growth. We note that the link prediction problem in a hybrid network is different from previously studied networks. Unlike the information networks and traditional online social networks, the structures in a hybrid network are more complicated and informative. We compare most popular and recent methods and principles for link prediction and recommendation. Finally we propose a novel structure-based personalized link prediction model and compare its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Experimentation,Performance

Keywords: link prediction, link analysis, Twitter

1. INTRODUCTION

The use of online social networks and social media in general has surged in recent years. In this work, we focus on the understanding of the use of one particular type of social service—that of the microblogging network. In microblog services such as Twitter, Yammer and Google Buzz, participants form an explicit social network by "following" (subscribing to) another user and thus automatically receive the (short) messages generated by the target user. Unlike some online social networks such as Facebook, LinkedIn or Myspace, a followed user has the option but not the requirement to similarly follow back. Thus, relationships in these social networks may be asymmetric, leading to three kinds of link relationships between users A and B. If A follows B, we say that A is a follower of B, and that B is a friend of A. If A and B both follow each other, we consider them mutual friends or reciprocal friends.

Thus, user B in a microblog service can generate messages, which are generally public and searchable, and any followers of B, such as A, will automatically receive those messages along with messages generated by all other users that A follows. The combination of multiple message intentions and asymmetry of connections has led some to call microblogging services such as Twitter "hybrid networks" [14, 22]. They are hybrid not just because they can carry multiple types of messages, but also because participants create links for multiple reasons—to be social (e.g., to connect online to existing offline social contacts) or to link to information sources. With multiple types of users, it may be difficult to understand how microblogging networks grow and evolve. In a hybrid social-information network, there are two viewpoints to consider. In an information network, the link prediction problem is like the recommendation problem, which is to recommend an information source to an information consumer. In a social network, the problem is to recommend friends to the users, as introduced by Liben-Nowell and Kleinberg [16]. If we can predict the next link that a user will likely create, we will 1) have a model of the user's interests that may be of value in recommending new links (e.g., as in Twitter's recently introduced "Who to follow" friend suggestions, and many third-party suggestion services) and in detecting communities; 2) be closer to modeling the network's overall growth processes; and, 3) be able to simplify the task of adding that link when the user wishes to do so.

In this paper, we analyze link structures in Twitter to predict future links. Our contributions are as follows. 1) We are the first to experimentally compare many popular link prediction methods in a microblogging network. Furthermore, we also compare with matrix factorization—a popular method of recommender systems. 2) We propose a novel structure-based approach to link prediction. Empirical results on ego-centric networks of Twitter users show that our method can outperform state-of-the-art methods on this task.

1.1 Related work

There are several fundamental kinds of link prediction methods, such as structural methods, random walk methods and supervised methods. Liben-Nowell et al. surveyed an array of methods for link prediction in online social networks [16, 17].

One branch of structural methods is based on the local structure, such as *common neighbors*, *Jaccard coefficient* and *Adamic/Adar* [1] which refines the simple counting features by weighting rarer features more heavily. The *preferential attachment* method supposes that the likelihood that a new edge involves node v is propor-

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

tional to $\Gamma(v)$, the number of neighbors of v. Based on global structure information, Clauset et al. [4] present a general technique for inferring hierarchical structure from network data and show that the existence of the hierarchy can simultaneously explain and quantitatively reproduce many commonly observed topological properties of networks.

Another approach utilizes random walk methods such as *Rooted PageRank* [16, 17] which is a variation of PageRank [21] that measures the stationary probability of each node in a random walk that returns to the root with some probability in each step. Weng et al. [24] try to identify influential users of micro-blogging services by using LDA to analyze user interests. Yin et al. propose a method which augments the original graph with attribute nodes, and then uses random walk to calculate link relevance [26, 27]. *SimRank* [8] recursively defines the similarity of two nodes and can also be interpreted in terms of a random walk. The most recent random walk-like method is *PropFlow* [18] which calculates the probability that a restricted random walk starting at node v_i ends at v_j in l steps. Katz [10] proposes a path-based method, which defines a measure that sums over the collection of all paths from v_i to v_j, and assigns more weight to shorter paths. Recently, Backstrom et al. [2] proposed a supervised random walk method which combines information from the network structure with node and edge level attributes. Supervised learning then adjusts the weights on different attributes to guide a random walk on the graph.

In supervised methods, the link prediction problem is usually considered as a classification problem. Such methods extract features from training data and can include both topological features (as in [9]) and node features. Hasan et al. [7] use different kinds of features, such as proximity features, aggregated features and topological features, and also compare different kinds of classifiers. More recently, Lichtenwalter et al. [18] examine important factors in the link prediction problem and present a classification framework which employs their *PropFlow* as a feature.

If you consider link prediction as a recommendation problem, a popular method is matrix factorization [13, 12, 11, 19] where the algorithms find hidden features for users and items by factorizing the observation matrix. However, those methods are designed for a user-item pair, and never before used for link prediction in social network.

There is other related research about link prediction [20, 23] and hybrid networks. Kwak et al. [14] find that the relationship of following and being followed on Twitter is not reciprocal, unlike most other social networking sites such as Myspace and Facebook. Romero and Kleinberg [22] also introduce the hybrid network concept and explore the directed closure process in Twitter. Recently, Golder et al. [6] discuss prediction specifically in Twitter. They analyze several principles for link prediction, such as shared interests, shared followers, and mutuality. They also discuss their user study results in [5].

2. LINK PREDICTION

Here we introduce our prediction framework based on link structures. In a hybrid social-information network, structures can reflect many scenarios that may be useful for capturing users' interests and predicting potential links. In Figure 1, we can see some examples of various structural meanings: a) User v_u may be interested in v_c, because other similar users with v_u are following v_c. b) User v_u may want to follow v_c, because they may be friends with each other in real life and they are willing to use microblog as social networks. c) User v_u may want to follow v_c, because v_u is following other users which are following v_i while v_c is also following v_i and they may share the same interests. With these three examples, we have already seen some meanings of structures. We wish to design a model to exhaust such structural information for predicting new links.

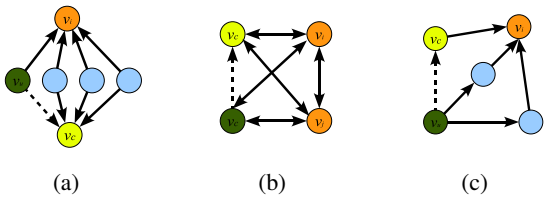


Figure 1: Examples of relationships between user v_u and candidate v_c.

Suppose that we want to recommend other users which user u may know or be interested in following. The problem we consider is that given a user u and the whole network G, what is the probability that user v_u follows user v_c: $P(v_u \to v_c|G)$. We will rank candidate users according to this equation, and the top N users will be recommended to user v_u. To calculate $P(v_u \to v_c|G)$, theoretically, each intermediate user/vertex v_i can contribute some structural information which represents two parts: the link structure between v_i and v_c and the link structure between v_u and v_i. Now let us define the set of target users to which we will recommend some friends V_u, the set of intermediate users which we will exhaust the structural information V_i, and the set of candidate users for recommendation V_c. Assuming that $P(v_u \to v_c|G)$ is the linear combination of all possible intermediate users/vertices' contribution, we have

$$P(v_u \to v_c|G) = \textstyle\sum_{v_i \in V_i} b_{v_i,v_c} \cdot a_{v_u,v_i}$$

Let b_{v_i,v_c} represent the contribution of the structural information between v_i and v_c, which can be considered as the strength of v_i's recommendation for v_c and a_{v_u,v_i} represent the contribution of the structural information between v_u and v_i, which can be considered as the score of v_u liking a recommendation of v_i. We will denote with A the matrix with elements $A_{v_u,v_i} = a_{v_u,v_i}$ and with A_{v_u} the column of A corresponding to v_u. Similarly, $B \in \mathcal{R}^{|V_i|\times|V_c|}$ with column vector B_{v_c}. Let $R_{v_u} = [r_{v_u,v_1}, r_{v_u,v_2}, ...r_{v_u,v_n}]$ represent the current friends snapshot of v_u in which $r_{v_u,v_i} = 1$ means v_i is a current friend of v_u and $r_{v_u,v_i} = 0$ means v_i is the current follower-only of v_u.

Elsewhere [25] we report that only 12% of follower-only users of all follower-only users become new friends; thus it is perhaps reasonable to use follower-only users as negative samples. Then,

$$\hat{R} = A^T B$$

In probabilistic view, we define the conditional distribution over the current friends.

$$p(R|A, B, \sigma^2) = \textstyle\prod_{v_u \in V_u} \prod_{v_c \in V_c} \left[\mathcal{N}(R|A_{v_u}^T B_{v_c}, \sigma^2)\right]^{I_{v_u,v_c}}$$

where $\mathcal{N}(R|A_{v_u}^T B_{v_c}, \sigma^2)$ is the probability function of the gaussian distribution with mean $A_{v_u}^T B_{v_c}$ and variance σ^2. I_{v_u,v_c} is the indicator function for selecting observed training data. For pair (v_u, v_c), if we use it as our training data, then $I_{v_u,v_c} = 1$, otherwise, $I_{v_u,v_c} = 0$. We also place zero-mean spherical Gaussian priors on the two structural parts A and B

$$p(A|\sigma_A^2) = \prod_{v_u \in V_u} \mathcal{N}(A_{v_u}|0, \sigma_A^2 \mathbf{I})$$

$$p(B|\sigma_B^2) = \prod_{v_c \in V_c} \mathcal{N}(B_{v_c}|0, \sigma_B^2 \mathbf{I})$$

The log of the posterior distribution over R,A and B is given by

$$\begin{aligned}
&\ln p(A,B|R,\sigma^2,\sigma_A^2,\sigma_b^2)\\
=\ &-\frac{1}{2\sigma^2}\sum_{v_u\in V_u}\sum_{v_c\in V_c} I_{v_u,v_c}(R_{v_u,v_c}-A_{v_u}^T B_{v_c})^2\\
&-\frac{1}{2\sigma_A^2}\sum_{v_u\in V_u} A_{v_u}^T A_{v_u}-\frac{1}{2\sigma_B^2}\sum_{v_c\in V_c} B_{v_c}^T B_{v_c}\\
&-\frac{1}{2}\left(\sum_{v_u\in V_u}\sum_{v_c\in V_c} I_{v_u,v_c}\right)\ln\sigma^2-\frac{1}{2}|V_u||V_i|\ln\sigma_A^2\\
&-\frac{1}{2}|V_c||V_i|\ln\sigma_B^2+C
\end{aligned}$$

where σ_A and σ_B control the smoothing factor of A and B. Let $\sigma_B=\sigma_A$, and then maximizing the log of the posterior distribution is equivalent to

$$\begin{aligned}
\min_{A,B}\quad &\sum_{v_u}\sum_{v_c} I_{v_u,v_c}(R_{v_u,v_c}-A_{v_u}^T B_{v_c})^2\\
&+\lambda_1(\|A\|_{Fro}^2+\|B\|_{Fro}^2)
\end{aligned}$$

where $\lambda_1=\sigma^2/\sigma_A^2$, is actually the smoothing factor and $\|\cdot\|_{Fro}^2$ denotes the Frobenius norm. Next, we need to involve structural regularization into the objective function.

2.1 Structural regularization

Elsewhere [25] we show that more than 90% of new links go to people two hops away from user (the ego). Intuitively, if two users v_i and v_j are far away on the graph, that is, the shortest path between v_i and v_j is too long, their structural information can be ignored. We can define the set of the effective structures S^e. For example, if we define that the structures with only one hop are effective, the set of effective structures will be $S^e=\{\Leftarrow,\Rightarrow,\Leftrightarrow\}$ and if we define that all structures with up to two hops are effective, then the set of effective structures will be $S^e=\{\Leftarrow,\Rightarrow,\Leftrightarrow,\Rightarrow\Rightarrow,\Rightarrow\Leftarrow,\Rightarrow\Leftrightarrow,\Leftarrow\Leftarrow,\Leftarrow\Rightarrow,\Leftarrow\Leftrightarrow,\Leftrightarrow\Leftarrow,\Leftrightarrow\Rightarrow,\Leftrightarrow\Leftrightarrow\}$. Let S_{v_i,v_j} represent the set of all possible structures from v_i to v_j and S_{v_i,v_j}^e represent the set of all effective structures from v_i to v_j—$S_{v_i,v_j}^e=S_{v_i,v_j}\cap S^e$. Thus, if $S_{v_u,v_i}^e=\emptyset$ where $v_u\in V_u$ and $v_i\in V_i$, then let $a_{v_u,v_i}=0$ and similarly, if $S_{v_i,v_c}^e=\emptyset$ where $v_i\in V_i$ and $v_c\in V_c$, then let $b_{v_i,v_c}=0$.

Beginning at some user $v_u\in V_u$, intuitively, if the structures of $(v_u \leftrightsquigarrow v_i)$ and $(v_u \leftrightsquigarrow v_j)$ are similar or same, the contribution scores of a_{v_u,v_i} and a_{v_u,v_j} should be similar. Following this intuition, we make constraints on structural scores matrix A, and define a structural regularization function $\mathcal{S}(A)$ to constrain similar scores on similar structures.

$$\mathcal{S}(A)=\frac{\sum_{v_u\in V_u}\sum_{v_i\in V_i}\sum_{v_j\in V_i}\mathcal{W}_{v_u}(v_i,v_j)(a_{v_u,v_i}-a_{v_u,v_j})^2}{\sum_{v_u\in V_u}\sum_{v_i\in V_i}\sum_{v_j\in V_i}\mathcal{W}_{v_u}(v_i,v_j)}$$

where $\mathcal{W}_{v_u}(v_i,v_j)$ is the measurement of similarity on structures attached on v_u: the more similar the structures of $(v_u \leftrightsquigarrow v_i)$ and $(v_u \leftrightsquigarrow v_j)$ are, the higher value the $\mathcal{W}_{v_u}(v_i,v_j)$ is. There are many kinds of methods to measure the structural similarity. Here, We list two:

Binary weighting if $S_{v_u,v_i}^e=S_{v_u,v_j}^e$, then $\mathcal{W}_{v_u}(v_i,v_j)=1$, otherwise $\mathcal{W}_{v_u}(v_i,v_j)=0$.

Cosine weighting let $N_{S_{v_u,v_i}^e}$ represent the vector of quantified effective structures of $(v_u \leftrightsquigarrow v_i)$, that is, $N_{S_{v_u,v_i}^e}=[n_{v_u\Rightarrow v_i},n_{v_u\Leftarrow v_i},n_{v_u\Leftrightarrow v_i},n_{v_u\Rightarrow\Rightarrow v_i}...]$, where $n_{v_u\Rightarrow\Rightarrow v_i}$ is the number of $\Rightarrow\Rightarrow$ path from v_u to v_i. Then, the cosine similarity is calculated as $\mathcal{W}_{v_u}(v_i,v_j)=\frac{N_{S_{v_u,v_i}^e}\cdot N_{S_{v_u,v_j}^e}}{\|N_{S_{v_u,v_i}^e}\|\cdot\|N_{S_{v_u,v_j}^e}\|}$

We also notice that if we take $S^e=\{\Leftarrow,\Rightarrow,\Leftrightarrow\}$, the two kinds of weighting are equivalent, because $n_{v_u\Rightarrow v_i}$, $n_{v_u\Leftarrow v_i}$ and $n_{v_u\Leftrightarrow v_i}$ only can be 0 or 1. Similarly, we add the structural constraints to B, and we have

$$\mathcal{S}(B)=\frac{\sum_{v_i\in V_i}\sum_{v_c\in V_c}\sum_{v_k\in V_c}\mathcal{W}_{v_i}(v_c,v_k)(b_{v_i,v_c}-b_{v_i,v_k})^2}{\sum_{v_i\in V_i}\sum_{v_c\in V_c}\sum_{v_k\in V_c}\mathcal{W}_{v_i}(v_c,v_k)}$$

The objective function $\mathcal{O}$ becomes

$$\begin{aligned}
\min_{A,B}\mathcal{O}\quad =\quad &\sum_{v_u\in V_u}\sum_{v_c\in V_c} I_{v_u,v_c}(R_{v_u,v_c}-A_{v_u}^T B_{v_c})^2\\
&+\lambda_1\|A_{v_u}\|_{Fro}^2+\lambda_1\|B\|_{Fro}^2\\
&+\lambda_2\mathcal{S}(A)+\lambda_2\mathcal{S}(B) \qquad (1)
\end{aligned}$$

where λ_2 is the structural factor tuning the weight of structural regularization. In the above model, we see the two parameters λ_1 controls the weight of smoothing and λ_2 controls the weight of regularization. The selected training links are represented by I_{v_u,v_c}.

2.2 Prediction in ego-centric networks

We call the above model the global model because the prediction is from the global network and performs collaborative filtering among all V_u. The global model will run on the whole graph to make predictions for a specific user and it will take a relatively long time to finish the computation; however, sometimes users perform interactive behaviors—such as requesting an instant recommendation. In this case, the global model may not work because of such long term computation. Secondly, the friendship network of some users may be already stable [25] and they may not want to add new friends.

It is necessary to make instant prediction for the users who are eager to get new friends. Unfortunately, directly reducing the model to fit the local structures of user v_u will cause overfitting. Thus, here we introduce a local model.

Considering the extreme case that only one user v_u requests new friends, the matrix R and A will reduce to only vectors R_{v_u} and A_{v_u} and a personalized method is necessary. We recall that the meaning of A and B, a_{v_u,v_i} can be considered as the probability of v_u trusting the recommendation of v_i and b_{v_i,v_c} can be considered as the probability of v_i recommending v_c. For b_{v_i,v_c}, because we know the current friendship network of v_u and also the structural information of v_i, we can make B personalized for v_u—B_{v_u}, that is, b_{v_u,v_i,v_c} means the probability of v_i recommending v_c for v_u, given the structure information of v_i. For some specific user v_u, we assume that v_u is interested in all his friends. Given the structure of some path between v_i and v_c $(v_i \leftrightsquigarrow v_c)$, we can use the following equation to get the approximation value of b_{v_u,v_i,v_c}:

$$\beta_{v_u,v_c,v_i}=\frac{\sum_{v_k\in V_{v_u\rightarrow}}\mathcal{W}_{v_i}(v_c,v_k)}{\sum_{v_k\in V}\mathcal{W}_{v_i}(v_c,v_k)}$$

where $v_u\in V_u$, $v_c\in V_c$ and $v_i\in V_i$. The above actually calculates the fraction of the number of v_u's friends who share similar structures with v_c over the number of all users who share similar structures with v_c. If the value β_{v_u,v_c,v_i} is larger, then there will be a larger probability that v_u will follow v_c. Then similarly as in

section 2, for some specific targeting user v_u we let

$$p(A_{v_u}|\sigma_A^2) = \mathcal{N}(A_{v_u}|0, \sigma_A^2\mathbf{I})$$
$$p(B|\beta_{v_u}, \sigma_K^2) = \prod_{v_c \in V_c} \mathcal{N}(B_{v_c}|\beta_{v_u,v_c}, \sigma_B^2\mathbf{I})$$

Then we have the objective function $\mathcal{O}_{v_u}$ for v_u:

$$\min_{A,B} \mathcal{O}_{v_u} = \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 + \lambda_1 \|A_{v_u}\|^2 + \lambda_1 \|B - \beta_u\|_{Fro}^2 + \lambda_2 \mathcal{S}(A_{v_u}) + \lambda_2 \mathcal{S}(B) \quad (2)$$

2.3 Solving the model

Solutions for Equations 1 and 2 are quite similar. One simple method is gradient descent. Intuitively, the structure rarely contributes negative effects and usually a user v_u likes some kinds of users or does not care about some other kinds of users. In the quantified observed matrix R, we also use 1 for current v_u's friends and 0 to represent links that v_u does not care about. From the Section 2.2, we also involve a guidance value—β. All these reasons lead us to constrain A and B to be nonnegative. Nonnegative matrix factorization has been researched for many years [15, 3].

The objective function $\mathcal{O}$ and $\mathcal{O}_{v_u}$ in Eq. 1 and Eq. 2 are not convex in both A and B together and it is realistic to expect an algorithm to find the global minima. The process we use for solving $\mathcal{O}$ and $\mathcal{O}_{v_u}$ in Equations 1 and 2 is to use an iterative algorithm following the methods in in [15, 3] to derive multiplicative update rules. The proof by Lee and Seung [15] suggests that the objective function will be nonincreasing under such update rules.

3. EXPERIMENTS

In this section, we describe our prediction experiments. Our method is only based on structural information of the social graph; thus for comparison methods, we also mainly focus on structure-based methods which do not involve user properties or content.

3.1 Data set and evaluation

In link prediction experiments, we use the same 979 Twitter users as in [25] and their immediate neighbors (979 ego users and their neighbors) that were collected to build a network for the link prediction task. In total, there are 211,559 unique users. For our experiments, we employed two kinds of evaluation methods.

Static Evaluation. Based on the 979 users' ego network snapshot on April 5th 2010, for each target user whose number of friends is larger than ten, we remove five links to friends. The prediction task is then to use the pruned networks to find the missing links. This evaluation method is widely used in the link prediction literature [4, 26, 27]. We use this process both for parameter tuning and for model analysis.

Dynamic Evaluation. We also monitored the changes in the 979 users' friendships and recorded the new links established between April 5th and May 12th [25]. Here, the prediction task is based on the April 5th network snapshot to predict new friends in the following months.

For validation purposes, we also run our experiments on a second static Twitter data set (described below in Section 3.6). Precision, recall and F-measure are calculated in the standard manner, and our main measurement is the F-measure based in the break even point.

3.2 Baselines

In this section, we analyze and discuss simple predictors and principles to show the difficulty of this problem. Daily monitored data shows that more than 90% of new users are within two hops and also their relationship [25]. Golder et al. [6, 5] discuss link prediction in Twitter, analyzing several principles for link prediction, such as shared interests, shared followers, and mutuality. Romero and Kleinberg [22] also introduce the directed closure process in Twitter tie formation. Here, we re-implement and compare the simple predictors which are from the principles described in [22, 6, 5].

Method	Static	Dynamic
Shared followers	0.078	0.119
Shared friends	0.061	0.083
Shared mutual	0.074	0.086
Common neighbors	0.071	0.116
Katz (l=2)	0.094	0.086

Table 1: Simple predictor analysis (F-measure).

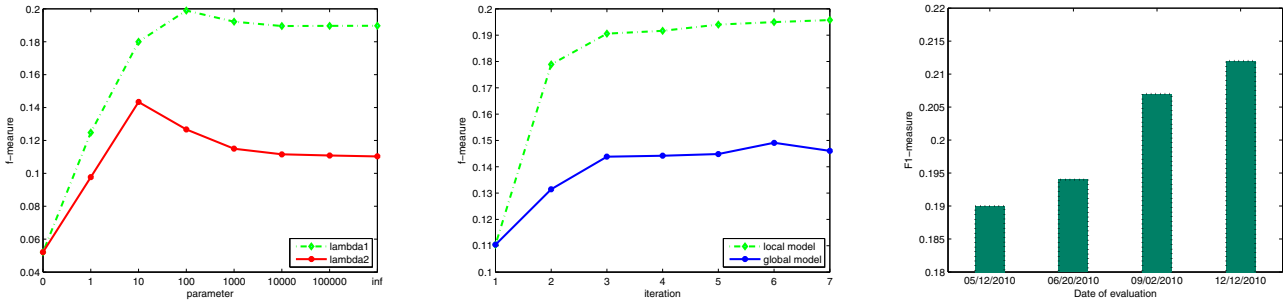
To represent the principle *Shared Interests*, we use the predictor: the number of shared friends. A shared interest is best represented by the relationship chain $v_u \rightarrow X \leftarrow v_c$. Similarly, *Shared Audience* ($v_u \leftarrow X \rightarrow v_c$) is measured by the number of shared followers. For *Transitivity* [6, 5] or the *Directed Closure Process* [22], we use Katz's methods with degree length $l = 2$, which is equivalent to the number of paths $v_u \rightarrow X \rightarrow v_c$. We also test *Shared Mutual Friends*. *Shared Neighbors* is just the count of the total number of neighbors (both friends and followers) without considering direction. The results are shown in Table 1.

From Table 1 we can see that all simple predictors provide similar performance—around .10 F-measure. We notice that the shared friends predictor performs worse than others, and that implies that two users sharing the same interests may not be particularly interested in following each other. Overall, simply using any single predictor cannot generate good results. Better methods are necessary.

3.3 Parameter analysis

In this section, we analyze our two models, and tune parameters on static data. In the experiments, we use the snapshots of the target user's friendship network to construct the observation matrix R: if user v_c is a friend of v_u, we will set the entry $r_{v_u,v_c} = 1$ and if user v_c is a follower-only of v_u, we will set the entry $r_{v_u,v_c} = 0$. Because we already know that more that 90% of new links are from second level neighbors, our effective structures are defined in one-hop; that is, each edge will have two parameters a and b respectively in A and B, and in the global model, it will generate prediction in two hops. In the local model, full structural information is captured in v_u's two-hop ego network. Initial values of A and B are all set to the same value. We finally find that when smoothing parameter $\lambda_1 = 100$ and regularization parameter $\lambda_2 = 100$ in the local model and $\lambda_1 = 1000$, $\lambda_2 = 100$ in the global model, the best performance is achieved. The model usually can converge within 10 iterations, and Figure 2(b) shows the performance changes as a function of iteration number. Because the current network is an ego-centric network which can provide full structural information, but the set of target users—V_u—is relatively small and may not provide good collaborative filtering, the performance of the local model is .197, which is better than the global model—.15. In the following, we use the local model for comparison.

Based on the local model and static data, we also analyze the effects of λ_1 and λ_2. Figure 2(a) shows the results. For the curve for λ_1, we set $\lambda_2 = 0$, and then tune λ_1 from 0 to infinity. We can see that it achieves the best performance when it is set to 100. We also note that when λ_2 is infinite, the model is reduced to the simple methods where links of the same type will share the same



(a) Parameter analysis:F-measure as a function of $\lambda_1(\lambda_2 = 0)$ and $\lambda_2(\lambda_1 = 0)$

(b) F-measure as a function of No. of iterations

(c) Evaluation on different time periods

Figure 2: Experimental analysis

value. The performance of this model is still competitive, although its F-measure is lower than the best performance. Similarly, the performance of tuning λ_2 is shown in the same figure.

3.4 Comparing to link prediction methods

In this section, we compare recent and popular methods which have been already used widely in link prediction problem. Six methods are implemented for comparison. The *Common neighbors* method simply counts the number of common neighbors. The *Jaccard coefficient* is calculated through dividing the number of common neighbors by the total number of neighbors. *Adamic/Adar* [1] refines the simple counting features by weighting rarer features more heavily. *Preferential attachment* scores are the product of v_c in-degree and v_u out-degree. We also compare the latest method—*PropFlow* [18]. In both Katz's method [10] and PropFlow, we tune the parameters l from 1 to 5 in static data.[1] Finally, we compare them on real dynamic data. The results are shown in the Static and Dynamic columns of Table 2.

In static evaluation, the results of *PropFlow*, *Common Neighbors*, *Jaccard Coefficient* and *Adamic/Adar* are similar and *PropFlow* which is a relatively newer method, gets better results than the other five competitors. *Jaccard Coefficient* shows competitive results which is similar with *PropFlow*. Preferential Attachment fails to predict missing links. For Preferential Attachment, because it only depends on the in-degree of the candidates, in the case of the information producers (with higher in-degrees), it may work. However, in real world, we know that individual users are more prevalent than information producers. Thus, we can imagine the failure of Preferential Attachment. In dynamic evaluation, a point which we have to note is that unlike in static evaluation, the *Jaccard coefficient* works very well and even better than *PropFlow*. Within ego-networks, the *Jaccard coefficient* is a competitive method and also simple to calculate. We also find the failure of *Preferential Attachment*. Our method outperforms all other methods in both static evaluation and dynamic evaluation.

3.5 Comparing to matrix factorization

As mentioned earlier, another direction to solve the link prediction problem in a hybrid network is to use the techniques of the traditional recommender systems. Matrix Factorization is a popular method which is widely used in recommender systems [12, 13]. Here we employ the standard matrix factorization methods with smoothing. The observation matrix R is the same as the one in our model and the objective function is as follows.

$$\min_{A,B} \sum_{v_u \in V_u} \sum_{v_c \in V_c} I_{v_u,v_c}(R_{v_u,v_c} - A_{v_u}^T B_{v_c})^2 + \lambda(\|A\|_{Fro}^2 + \|B\|_{Fro}^2)$$

To solve this optimization, we used stochastic gradient descent. Based on the static data, we tune the number of hidden features from 20 to 300, find the optimal parameter for comparison and set $\lambda = 0.05$. The results are shown in the last row of Table 2. On the static data, matrix factorization only achieves around .09 F-measure but the performance of matrix factorization on real data is also competitive at .163. Our model can outperform the standard matrix factorization in both static data and dynamic data because our method essentially incorporates matrix factorization techniques with structural information.

3.6 Validating results

To test for sensitivity to our data set and sampling methods, we also ran our experiments on a subset of the large Twitter follow graph collected by Kwak et al. at KAIST [14]. We randomly sample 2,000 test users and extract their ego networks as in Section 3.3. There are, in total, 81,580 users and almost 10 million edges within this second test network. We again compared our methods with the other seven methods using the static dataset methodology. The results are shown in the rightmost column of Table 2 and are consistent with our earlier experiments. PropFlow is also better than other comparison methods. Our approach consistently outperforms all other tested methods.

Method	Static	Dynamic	KAIST
Our model	**0.197**	**0.190**	**0.127**
PropFlow	0.124	0.099	0.081
Katz	0.094	0.086	0.077
Jaccard coefficient	0.098	0.169	0.079
Adamic/Adar	0.090	0.128	0.069
Common neighbors	0.071	0.116	0.051
Pref. Attachment	0.012	0.012	0.023
Matrix factorization	0.082	0.163	0.074

Table 2: Comparing link prediction methods (F-measure).

[1] In their paper, they also proposed a supervised method. Here, we select PropFlow for two reasons: First, for Lichtenwalter et al.'s supervised methods, there are many parameters to tune and selecting features is also a problem. Second, in their paper, PropFlow is used as a feature, and for most supervised methods, our method can also be used as a feature.

3.7 Discussion

We have demonstrated many of the challenges of link prediction in a hybrid network and also noticed that the overall performance is relatively low, compared to results presented in some link prediction papers on other datasets. However, even when considering "social networks", most existing work does not directly examine online social networks, but rather networks of co-authorship or similarly constructed networks reflecting some social relationship or record of activity.

On the other hand, the links in an online social network may reflect relationships (friends, family) that are not visible in a record of activity, and in a microblogging network with hybrid characteristics is even more complex.

As a result, previous methods which may work well on traditional social networks or co-authorship networks may not work as well on hybrid networks. Our results shows that the F-measure of many popular methods on our real-world data is only around 0.10.

Another cause for low performance of link prediction is that the microblogging network continues to grow. Each day, there may be many new links created [25]. In our experiments, we only evaluate new links within the following one month, so performance may be underestimated. It is possible that users are actually interested in those predicted links but they may not create those links within the following one month due to the fact that users may not discover those potential friends in a short period of time. In other words, users may create those links later, after our initial evaluation period. We conduct a simple experiment to test this: we make predictions based on the same training data—the 04/05/2010 snapshot, but we evaluate on different snapshots from different times. Figure 2(c) shows the results, and we find that after 05/12/2010, target users continue to create links which we had predicted, so measured performance grows higher and higher.

Another thing we can notice is that performance on the three test sets are different. For example, matrix factorization works well on the dynamic data but not well on the static data. We can imagine that static evaluation and dynamic evaluation have different properties such that some methods are better suited for one or the other. For the prediction task, dynamic evaluation is a more accurate estimate of future performance than static evaluation. However, if recommendation is the true end goal, it is difficult to tell which (if any) is better without involving a user study.

4. SUMMARY

In this paper, we examined the link structure and link prediction task within the Twitter microblogging network. We proposed a novel personalized structure-based link prediction model and compared its predictive performance against many fundamental and popular link prediction methods on real-world data from the Twitter microblogging network. Our experiments on both static and dynamic data sets show that our methods noticeably outperform the state-of-the-art.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Number IIS-0545875. We thank Roger Nagel, Xiong Xiong, Xiaoguang Qi, Lifeng Shang and Yong Zhang for their suggestions. Finally, we thank Twitter for access to their service and Kwak et al. [14] for releasing their data.

5. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proc. WSDM*, 635-644, 2011.

[3] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *Proc. IJCAI*, pages 1010-1015, 2009.

[4] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98-101, May 2008.

[5] S. Golder and S. Yardi. Structural predictors of tie formation in Twitter: Transitivity and mutuality. In *Proc. SocialCom*, pages 88-95, 2010.

[6] S. Golder, S. Yardi, M. Marwick, and d. boyd. A structural approach to contact recommendations in online social networks. In *Proc. 2nd Workshop on Search in Social Media (SSM)*, 2009.

[7] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Proc. of SDM workshop on Link Analysis, Counterterrorism and Security*, 2006.

[8] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *Proc. KDD*, pages 538-543, 2002.

[9] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *Proc. ICDM*, pages 340-349, 2006.

[10] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39-43, March 1953.

[11] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. KDD*, pages 426-434, 2008.

[12] Y. Koren. Collaborative filtering with temporal dynamics. In *Proc. KDD*, pages 447-456, 2009.

[13] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30-37, 2009.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proc. WWW*, pages 591-600, 2010.

[15] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. NIPS*, pages 556-562, 2001.

[16] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proc. CIKM*, pages 556-559, 2003.

[17] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007.

[18] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proc. KDD*, 2010.

[19] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Proc. NIPS*, 2003.

[20] T. Murata and S. Moriyasu. Link prediction of social networks based on weighted proximity measures. In *Proc. WI*, pages 85-88, 2007.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[22] D. Romero and J. Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *Proc. ICWSM*, 2010.

[23] T. Tylenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. In *Proc. 3rd Workshop on Soc. Net. Mining and Analysis (SNA-KDD)*, pages 1-10, 2009.

[24] J. Weng, E.-P. Lim, J. Jiang, and Q. He. TwitterRank: finding topic-sensitive influential twitterers. In *Proc. WSDM*, pages 261-270, 2010.

[25] D. Yin, L. Hong, X. Xiong, and B. D. Davison. Link formation analysis in microblogs. In *Proc. SIGIR*, pages 1235-1236, 2011.

[26] Z. Yin, M. Gupta, T. Weninger, and J. Han. LINKREC: a unified framework for link recommendation with user attributes and graph structure. In *Proc. WWW*, pages 1211-1212, 2010.

[27] Z. Yin, M. Gupta, T. Weninger, and J. Han. A unified framework for link recommendation using random walks. In *Proc. of the Int'l Conf. on Adv. in Social Net. Analysis and Mining (ASONAM)*, pages 152-159, 2010.