Supervised Link Prediction Using Multiple Sources

Zhengdong Lu[*] Berkant Savas[*] Wei Tang[†] Inderjit Dhillon[†]

October 10, 2010

The University of Texas at Austin

Department of Computer Science

Technical report TR–10–35

Abstract

Link prediction is a fundamental problem in social network analysis and modern-day commercial applications such as Facebook and Myspace. Most existing research approaches this problem by exploring the topological structure of a social network using only one source of information in an unsupervised and heuristic manner. However, in many application domains, in addition to the social network of interest, there are a number of auxiliary social networks and/or derived proximity networks available. In this paper we propose a general framework of supervised link prediction from multiple heterogeneous sources. The contribution of the paper is twofold: (1) a supervised learning framework that can effectively and efficiently learn the dynamics of social networks in the presence of auxiliary networks; (2) a feature design scheme for constructing a rich variety of path-based features using multiple sources, and an effective feature selection strategy based on structured sparsity. Extensive experiments on three real-world collaboration networks show that our model can effectively learn to predict new links using multiple sources, yielding higher prediction accuracy than unsupervised and single-source supervised models.

Keywords: social network; link prediction; multiple sources; supervised learning;

1 Introduction

Social networks are dynamic by nature. They change quickly over time when new relationships establish between people (called *actors*), and when old relationships dissolve. These relational changes (when friends of friends become friends), characteristics of the actors (actor covariates), characteristics of pairs of actors (dyadic covariates), and random unexplained influences are the joint contribution to the dynamics of a network topology. Understanding the mechanisms by which the social networks evolve is a fundamental question that is still not well understood, and it forms the motivation for our work here.

In addition to the links in the network, we may also have exogenous features with various level of uncertainty, most interestingly the auxiliary networks between the same group of vertices from heterogeneous sources. Take the Facebook network for example, besides the friendship relations between the users, there are other relations based on blog article citations and commenting, or online messaging. Another example is the so-called collaboration network among scientific researchers. A collaboration relation forms between two researchers if they have co-authored a paper, but there are other types of relations or proximity that are informative for telling whether they will have collaboration in the future, e.g., whether they have attended

[*] Institute for Computational Engineering and Sciences, The University of Texas at Austin.
[†] Department of Computer Science, The University of Texas at Austin.

the same conference, whether they have cited the same papers, or whether they have published papers with similar keywords.

In this paper, we focus on exploiting the topological information for a basic computational problem underlying social network evolution—the link prediction problem. The setting is: given snapshots of an evolving social network from time 1 to t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time $t+1$. This problem is also related to uncovering hidden links in a network, which can be considered as a missing value problem for entries of the graph's adjacency matrix. Various unsupervised [1, 15, 19, 21] and supervised [5, 8, 13, 17, 26] models have been proposed to address these problems, assuming there is one network available. However, there is little work on incorporating auxiliary sources in link prediction. Kashima et al. [14] introduced a *link propagation* framework to exploit multiple types of links between vertices. However, this work is largely unsupervised, and only works for missing link recovery in static networks. In constrast, we aim to find a predictive model for evolving networks and learn the dynamics with a supervised framework.

Another thread of existing work attempts to understand the mechanism underlying the evolving social network from a time series of network snapshots. The study in that direction usually focuses largely on (1) extracting basic trends of social network evolution, such as stability, reciprocity, and transitivity [5,6,23,24], and (2) understanding how simple local dynamics give rise to the global structure of a social network, such as decreasing diameters (related to the small world effect) and power law distribution of vertex degrees [18,21]. These models are designed mainly to understand various statistical aspects of the observed social networks, and usually lack the power of predicting future links.

Overview

We intend to marry the rigorous treatment in statistical network models, and the effectiveness of ad hoc and unsupervised link prediction algorithms. The two threads of research are briefly reviewed in Section 2. Then in Section 3, we elaborate on the dynamic model for social network evolution in the presence of multiple auxiliary networks. Using an exponential family distribution to describe the dynamics, we reduce the learning of dynamics to logistic regression models, with features summarizing the local properties of the vertex pairs. In Section 4.1, we show how this dynamic model can be used to learn from historical snapshots and predict future links. Our path-counting feature design can effectively encode the topological information from auxiliary networks, and yield a rich set of features. Then in Section 5, we discuss effective strategies to trim these features with supervision. In Section 6 we test our algorithms on three real life collaboration data sets: arXiv-HepTh, SIAM, and CiteSeer. The results show that the proposed method can effectively learn the dynamical aspects essential for prediction, and outperforms its unsupervised counterparts.

Our contribution in this paper is twofold:

1. a supervised learning framework that can effectively and efficiently learn the dynamics of social networks in the presence of auxiliary networks;
2. a feature design scheme for constructing a rich variety of path-based features using multiple sources, and an effective feature selection strategy based on structured sparsity.

Notation

We use bold upper-case letters, e.g., $\mathbf{A}$, for matrices, bold lower-case letters, e.g., $\mathbf{b}$, for vectors, and italics for scalars, e.g., a. We use superscripts in parenthesis for time steps, e.g., $\mathbf{A}^{(t)}$, and power without parenthesis, e.g., $\mathbf{A}^n$. Also when denoting one set of objects with consecutive time indices from t_1 to t_2, we use $\mathbf{A}^{(t_1:t_2)}$. We save subscripts for indexing entries in a matrix or a vector, e.g., A_{ij}, and θ_i.

2 Background

Our work is motivated by the long thread of work in statistical modeling of static and dynamical social networks, as well as the work on heuristic but practically effective unsupervised link prediction models. We now give a brief introduction to the two contrasting threads of work, with emphasis on the parts that are directly related to our model.

2.1 Unsupervised Link Prediction

Various models have been proposed for link prediction, which, as summarized in [19], generally fall into three categories. The first category has methods based on vertex neighborhoods, including Common neighbors [21], Adamic/Adar [1], Preferential Attachment [3,20]. The second category has methods based on the ensemble of all paths, including Katz [15] and Hitting Time, while the third category includes high level approaches, such as matrix factorization and clustering. All of these methods rely on a predictive score function for all entries to get a ranking of edges that are likely to occur.

We will elaborate on the Katz measure [15], for its modeling simplicity and its wide success in practice. More importantly, as we will show later, Katz is closely related to the proposed framework and provides a justification for our work. The Katz directly sums over the collection of paths, exponentially damped by length to count short paths more heavily, leading to the β-parametrized measure:

$$\mathsf{score}_{\mathsf{Katz}}(i,j) = \sum_{l=1}^{\infty} \beta^l |\mathsf{path}_{i,j}^{\langle l \rangle}|, \tag{1}$$

where $\mathsf{path}_{i,j}^{\langle l \rangle}$ is the set of all length-l paths from vertex i to j. With $\mathbf{A}$ being the adjacency matrix of the graph, one can verify that for $\beta < 1/\|\mathbf{A}\|_2$, the score matrix is given by

$$\mathsf{score}_{\mathsf{Katz}} = \sum_{l=1}^{\infty} \beta^l \mathbf{A}^l = (\mathbf{I} - \beta\mathbf{A})^{-1} - \mathbf{I}. \tag{2}$$

When inverting $(\mathbf{I} - \beta\mathbf{A})$ becomes too expensive, one can choose to stop after paths of length l_{max} in (2) to get the truncated Katz score:

$$\mathsf{score}_{\mathsf{tKatz}} = \sum_{l=1}^{l_{max}} \beta^l \mathbf{A}^l. \tag{3}$$

It is easy to see that truncated Katz becomes a good approximation of Katz when β is small enough. In practice truncated Katz often outperforms Katz for link prediction.

There are many other predictive models in the same spirit as Katz, with a different way of calculating the proximity score (see [19] for a comprehensive survey):

- Common Neighbors: $\mathsf{score}_{\mathrm{CN}}(i,j) = |\Gamma(i) \cap \Gamma(j)|$, where $\Gamma(i)$ denotes all vertex i's neighbors. It is easy to verify that the $\mathsf{score}_{\mathsf{Katz}}(i,j)$ with small enough β yields predictions on new links much like common neighbors, since paths of length (≥ 2) will be damped and contribute very little to the summation;
- Preferential Attachment: the product of the degrees of two vertices, $\mathsf{score}_{\mathsf{PA}}(i,j) = |\Gamma(i)| \cdot |\Gamma(j)|$;
- Adamic/Adar: a weighted version of common neighbors, $\mathsf{score}_{\mathsf{AA}}(i,j) = \sum_{k \in \Gamma(i) \cap \Gamma(j)} \frac{1}{\log |\Gamma(k)|}$;
- Graph Length: negated length of the shortest path between i and j.

2.2 Dynamical Random Graph Models

In contrast to the heuristic methods for link prediction, there is a group of models devoted to study the intrinsic mechanism governing the topological changes of networks over time. For a series of snapshots $\mathbf{A}^{(\tau)}$ of a network at different time steps,

$$\cdots \to \mathbf{A}^{(\tau-2)} \to \mathbf{A}^{(\tau-1)} \to \mathbf{A}^{(\tau)} \to \mathbf{A}^{(\tau+1)} \to \cdots,$$

a statistical model for network evolution can be estimated. Usually it is assumed that the underlying states of the social network follow a stationary Markov process, and the statistical modeling therefore boils down to the modeling of a transition probability, $\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)})$. For example, Snijders [23] has proposed a continuous time model of network dynamics, where each observed event represents a single actor altering his or her outgoing links to optimize an objective function based on local neighborhood statistics. Robins and Pattison [22] and Hanneke et al. [6] have studied a family of models of network dynamics over discrete time steps, with an exponential random graph model (ERGM) describing the transition probability

$$\mathcal{P}(\mathbf{A}^{(t+1)}|\mathbf{A}^{(t)}) \propto e^{\langle \boldsymbol{\theta}, \Phi(\mathbf{A}^{(t+1)}|\mathbf{A}^{(t)})\rangle},$$

where $\Phi(\mathbf{A}^{(t+1)}|\mathbf{A}^{(t)})$ denotes the vector of sufficient statistics and $\boldsymbol{\theta}$ denotes the natural parameters. The inference problem with this model is in general intractable (unless other simplifying assumptions are made), and requires approximation methods like sampling. Typically, these models are concerned with dynamical properties such as stability, reciprocity (for directed graph), and transitivity. Although this line of work has brought up a rich pool of features and statistical aspects that are potentially useful for link prediction, the models are not tailored for prediction and certainly lack the efficiency for a real-world prediction task.

3 Learning The Dynamics

3.1 Dynamics of Social Network Evolutions

In this section we will describe in detail our model for the dynamics of social network evolution in the presence of multiple auxiliary networks. For simplicity, we only consider the undirected unweighted graph, which implies that all relationships are mutual and weighted equally. It is straightforward to extend these models to directed and/or weighted graphs.

Suppose we observe snapshots of an evolving social network from time 1 to time t, with the corresponding adjacency matrices denoted $\mathbf{A}^{(1)}$ through $\mathbf{A}^{(t)}$. The task is to find a prediction model for $\mathbf{A}^{(t+1)}$. We assume no vertices in $\mathbf{A}^{(\tau)}, \tau = 1, \cdots, t$, are added or removed during the evolution, but edges could form and/or disappear at each time step. In addition to our observations on $\{\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(t)}\}$, we also have available snapshots of a network from heterogeneous but related sources denoted $\{\mathbf{B}^{(1)}, \cdots, \mathbf{B}^{(t)}\}$. Extension to more than one auxiliary network is straightforward, but is omitted here for description simplicity.

We start describing our model with the following two assumptions:

Assumption I: The evolution of $\mathbf{A}^{(\tau)}, \tau = 1, \cdots, t+1$, is a Markov process, where the probability of network state $\mathbf{A}^{(\tau)}$ is governed jointly by $\mathbf{A}^{(\tau-1)}$ and $\mathbf{B}^{(\tau-1)}$. :

$$\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(1:\tau-1)}, \mathbf{B}^{(1:\tau-1)}) = \mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}),$$

as illustrated in Figure 1.

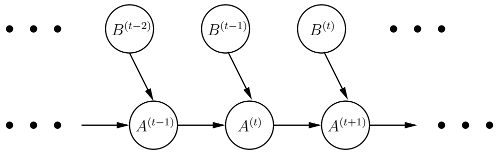

Figure 1: Illustration of a hybrid Markov process.

Assumption II: $\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ fully factorizes:

$$\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) = \prod_{i,j} \mathcal{P}(A_{ij}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}).$$

Both assumptions are made for modeling tractability. Assumption I is made in practically all models for social network dynamics, starting from Katz and Proctors' discrete Markov chain model [16]. This assumption is usually not realistic, but leads to more manageable models. We can loosen Assumption I to include the dependence of current snapshot on a longer history,

$$\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(1:\tau-1)}, \mathbf{B}^{(1:\tau-1)}) = \mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-m:\tau-1)}, \mathbf{B}^{(\tau-m:\tau-1)}),$$

which makes more sense in a collaboration network, since the underlying relationship may not appear as observable events (e.g., co-authoring a paper) in the duration time of a certain snapshot. This is in contrast to on-line social networks such as Facebook, where links (friendship) are much more persistent and stable, therefore the most recent snapshots usually contain almost all the information needed for prediction. For notational simplicity, we will describe the case for $m = 1$, while discussing the case of multiple retrospective steps only when the extension is not trivial.

Assumption II is necessary because of the difficulty in estimating the transition probability $\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$. To simplify, we assume the independence of link formation in $\mathbf{A}^{(\tau)}$ conditional on the history at time $\tau - 1$. In practice, when each time step covers a reasonably long duration, this assumption may be violated. For example, a new link could form between vertices i and j at time $\tau - 1 + \Delta\tau$ ($\Delta\tau < 1$) because of two other new links (i, k) and (k, j) formed before time $\tau - 1 + \Delta\tau$ but after $\tau - 1$, and hence these three links are not independent of each other. Nevertheless, this assumption greatly reduces the modeling complexity, and works well in practice.

3.2 Probabilistic Model

We generalize the exponential random graph model (ERGM) [6, 22] to describe the transition probability

$$\mathcal{P}(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) \propto e^{\langle \boldsymbol{\theta}, \Phi(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})\rangle},$$

where $\Phi(\mathbf{A}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ is the "sufficient statistics" associated with $\mathbf{A}^{(\tau)}$ conditioned on the historical states $\mathbf{A}^{(\tau-1)}$ and $\mathbf{B}^{(\tau-1)}$, and $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_K]^\top$ denotes the natural parameters to be learned. From Assumption II, the transition probability can be further simplified to a fully factorized exponential family distribution, with the probability for each link $\mathcal{P}(A_{ij}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ modeled as

$$\frac{1}{Z_{ij}(\boldsymbol{\theta}, \mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})} e^{\sum_{k=1}^K \theta_k \phi_k(A_{ij}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})},$$

where $\phi_k(A_{ij}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ is the k^{th} statistic associated with pair (i, j), and $Z_{ij}(\boldsymbol{\theta}, \mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ is the normalization constant (partition function). Since we are modeling the presence/absence of the link $A_{ij}^{(\tau)}$, one natural choice of the feature ϕ_k is

$$\phi_k(A_{ij}^{(\tau)}|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) = A_{ij}^{(\tau)} \cdot g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) \tag{4}$$

where $A_{ij}^{(\tau)} \in \{0, 1\}$, and $g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ is the k^{th} feature extracted from previous snapshot $\mathbf{A}^{(\tau-1)}$ and $\mathbf{B}^{(\tau-1)}$ for pair (i, j). Usually $g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ summarizes a certain local property from $\mathbf{A}^{(\tau-1)}$ and $\mathbf{B}^{(\tau-1)}$ of interest to the generation of link (i, j), e.g. the number of common neighbors in $\mathbf{B}^{(\tau-1)}$

$$g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) = \sum_n B_{in}^{(\tau-1)} B_{nj}^{(\tau-1)},$$

which is actually a special case of the path-counting feature we will introduce in Section 4.3. It is critical to note that we assume the anonymity of all vertices, and consider that all the links are formed based on the same local dynamics. This implies the same parameter θ_k for each $g_{k,ij}$ for all (i, j), rendering the learning of $\boldsymbol{\theta}$ feasible.

It follows from (4) that the probabilistic model is a logistic regression with

$$\mathcal{P}(A_{ij}^{(\tau)} = 1|\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) = \frac{e^{\sum_{k=1}^K \theta_k g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})}}{1 + e^{\sum_{k=1}^K \theta_k g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})}}. \tag{5}$$

This implies that the probability of having a link formed beteween i and j at time τ is governed by a latent potential

$$p_{\boldsymbol{\theta}}(i,j) = \sum_{k=1}^{K} \theta_k g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}), \tag{6}$$

which is a linear combination of features $g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)})$ from snapshot at time $\tau - 1$.

4 Model Fitting

Suppose we want to predict the links in snapshot $\mathbf{A}^{(t+1)}$, and have as observations the historical snapshots of the main network $\{\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(t)}\}$ as well as auxiliary network $\{\mathbf{B}^{(1)}, \cdots, \mathbf{B}^{(t)}\}$. Extension to more than one auxiliary network is straightforward, and is omitted here for simplicity.

4.1 Model Fitting and Prediction

It is established in Section 3 that the generative model for links is essentially logistic regression with unknown parameters $\boldsymbol{\theta}$. The task of model fitting is therefore to determine $\boldsymbol{\theta}$ from the observation $\{\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(t)}\}$ and $\{\mathbf{B}^{(1)}, \cdots, \mathbf{B}^{(t)}\}$, and predict the links in $\mathbf{A}^{(t+1)}$. The problem we focus on is the formation of *new* links in the main network, i.e. links that do not appear in the retrospective steps. Let $\mathcal{E}_\tau$ denote the set of links in snapshot τ. Let $\mathcal{N}_\tau$ denote the new links formed in time interval $[\tau, \tau+1]$ and $\mathcal{Z}_\tau$ denote the complement of $\mathcal{E}_\tau \cup \mathcal{N}_\tau$. Clearly $\mathcal{E}_{\tau+1} = \mathcal{E}_\tau \cup \mathcal{N}_\tau$ while $\mathcal{E}_\tau \cup \mathcal{N}_\tau \cup \mathcal{Z}_\tau$ is the set of all possible pairs. The model fitting task is to find the parameters $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_n]^\top$ that maximize the likelihood of the observed new links from time step 2 to time step t

$$\boldsymbol{\theta}^* \quad = \quad \arg\max_{\boldsymbol{\theta}} \prod_{\tau=2}^{t} \prod_{i,j \in \mathcal{N}_\tau \cup \mathcal{Z}_\tau} \mathcal{P}(A_{ij}^{(\tau)} | \mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}).$$

With the probability model we described in (5), the negative log likelihood we minimize is

$$L(\boldsymbol{\theta}) = -\sum_{\tau=2}^{t} \Bigg\{ \sum_{i,j \in \mathcal{N}_\tau \cup \mathcal{Z}_\tau} \sum_{k=1}^{K} \theta_k g_{k,ij}(\mathbf{A}^{(\tau-1)}, \mathbf{B}^{(\tau-1)}) - \tag{7}$$

$$\sum_{i,j \in \mathcal{Z}_\tau \cup \mathcal{N}_\tau} \log(1 + e^{\sum_{k=1}^{K} \theta_k g_{k,ij}(\mathbf{A}^{(\tau-1)},, \mathbf{B}^{(\tau-1)})}) \Bigg\},$$

which is convex in $\boldsymbol{\theta}$ and various optimization routines can be used to get a global minimum.

Once the optimal parameter $\boldsymbol{\theta}^*$ is obtained, the prediction of $\mathbf{A}^{(t+1)}$ can be carried out using the poential in (6) as the score

$$\mathsf{score}_{\boldsymbol{\theta}^*}(i,j) = p_{\boldsymbol{\theta}^*}(i,j) = \sum_{k} \theta_k^* g_{k,ij}(\mathbf{A}^{(t)}, \mathbf{B}^{(t)}), \tag{8}$$

which can also be justified since $\mathsf{score}_{\boldsymbol{\theta}^*}(i,j)$ is also the log odds ratio $\log \frac{\mathcal{P}(A_{ij}^{(t+1)}=1|\mathbf{A}^{(t)},\mathbf{B}^{(t)})}{\mathcal{P}(A_{ij}^{(t+1)}=0|\mathbf{A}^{(t)},\mathbf{B}^{(t)})}$. We use the score function (instead of the actual probability) in link prediction if only the ranking of the predicted links are needed, since it is clear that the score preserves the ordering of the likelihood

$$\mathcal{P}(A_{ij}^{(t+1)} = 1 | \mathbf{A}^{(t)}, \mathbf{B}^{(t)}) > \mathcal{P}(A_{i'j'}^{(t+1)} = 1 | \mathbf{A}^{(t)}, \mathbf{B}^{(t)})$$

$$\Longleftrightarrow \mathsf{score}_{\boldsymbol{\theta}^*}(i,j) > \mathsf{score}_{\boldsymbol{\theta}^*}(i',j'). \tag{9}$$

4.2 Square Loss Surrogate

The logistic regression model is still computationally expensive for many real-world applications. Here we show that the simple square loss can be used as a cheap and effective surrogate for the logistic regression objective, with detailed analysis on time complexity given in Section 4.5.

It is easy to see that the potential $p_{\boldsymbol{\theta}}(i,j)$ in (6) is positive when the probability of $A_{ij}^{(t)} = 1$ ("link") is greater than $A_{ij}^{(t)} = 0$ ("no link"), and vice versa. A simple heuristic of fitting the scores of "linked" pairs to $+1$ and "not-linked" pairs to -1 leads to the quadratic surrogate objective function for $\boldsymbol{\theta}$

$$L_{\text{lsq}}(\boldsymbol{\theta}) = \sum_{i,j\in\mathcal{N}_t\cup\mathcal{Z}_t} \|p_{\boldsymbol{\theta}}(i,j) - \text{sign}(A_{ij}^{(t)} - 0.5)\|^2 \tag{10}$$

$$= \sum_{i,j\in\mathcal{N}_t\cup\mathcal{Z}_t} \Big(\sum_k \theta_k g_{k,ij}(\mathbf{A}^{(t)}, \mathbf{B}^{(t)}) - \text{sign}(A_{ij}^{(t)} - 0.5) \Big)^2 \tag{11}$$

where sign$(\cdot)$ returns the sign ($+1$ or -1) of the input argument. Equation (11) can be rearranged into the following matrix form

$$L_{\text{lsq}}(\boldsymbol{\theta}) = \|\mathbf{S}\boldsymbol{\theta} - \mathbf{y}\|_2^2, \tag{12}$$

where the number of rows in $\mathbf{S}$ is $|\mathcal{N}_t \cup \mathcal{Z}_t|$ and $\mathbf{y}$ is a target vector. Note that the model with the square loss surrogate differs the original model only in the training phase. Once the model parameter $\boldsymbol{\theta}$ is obtained, the testing phase (prediction) is identical for the two objectives, since they have the same formula for scores.

4.3 Path-counting Features

The features $g_{k,ij}(\mathbf{A}^{(\tau)}, \mathbf{B}^{(\tau)})$ in (4) could take a great variety of knowledge about the possible links between vertices i and j. For example, this could be based on information about vertex i and j, e.g. the demographical data about them, or it could be from topological properties of the network. Examples of topology-based features include those associated with other unsupervised link prediction models, e.g. Preferential Attachment and Adamic/Adar, and measurements used to characterize graph topology, such as clustering coefficients [10,21]. In this paper, we are particularly interested in path-counting features, since it has been shown to be a simple but informative measure of proximity between vertices. Also, as we shall show, our supervised model with path-counting features are natural extension to popular unsupervised models such as Katz measure (and hence nearest neighbors).

The path-counting features for a single graph/network source are simply the number of length-l paths with $l = 1, \cdots, l_{max}$. With any unweighted graph the l^{th} feature on any snapshot τ can be computed from the adjacency matrix

$$g_{l,ij}(\mathbf{A}^{(\tau)}) = |\mathsf{path}_{i,j}^{\langle l\rangle}| = \sum_{n_1,\ldots,n_{l-1}} A_{in_1}^{(\tau)} A_{n_1 n_2}^{(\tau)} \cdots A_{n_{l-1}j}^{(\tau)}, \tag{13}$$

while (13) is also used for weighted graph in this paper. The same feature is much more concise in matrix form. For example, the same l^{th} feature for all pairs (i,j) is simply $\mathbf{G}_l = (\mathbf{A}^{(\tau)})^l$. It is easy to verify that the features corresponding to paths with length $0, 1, \cdots, l_{max}$ are given by terms in the matrix polynomial

$$\mathbf{G}_l = (\mathbf{A}^{(\tau)} + \mathbf{I})^l. \tag{14}$$

where the identity matrix $\mathbf{I}$ is for the "length 0" paths, which also servers as a constant feature, or offset in the logistic regression.

With multiple sources, we will have a much richer set of paths if we allow cross routes between networks from different sources. The best way to understand this is through the concept of a *multigraph* [7], which allows more than one edge between two vertices. Suppose we have a multigraph $\mathcal{M}$, between any two vertices there could be an edge from $\mathbf{A}^{(\tau)}$ and an edge from $\mathbf{B}^{(\tau)}$. For description convenience, we can have the two kind of edges color-coded, "A" colored versus "B" colored. This results in three types of paths in $\mathcal{M}$:

1. Pure color paths with only edges of $\mathbf{A}$, e.g., $i \xrightarrow{\mathbf{A}} j \xrightarrow{\mathbf{A}} k$;

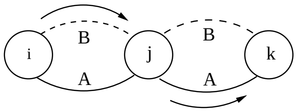

Figure 2: Example of a hybrid color path, $i \xrightarrow{\mathbf{B}} j \xrightarrow{\mathbf{A}} k$

2. Pure color paths with only edges of $\mathbf{B}$, e.g., $i \xrightarrow{\mathbf{B}} j \xrightarrow{\mathbf{B}} k$;
3. Hybrid color paths with edges of both $\mathbf{A}$ and $\mathbf{B}$, e.g., $i \xrightarrow{\mathbf{B}} j \xrightarrow{\mathbf{A}} k$, as illustrated in Figure 2.

The counting of the type-1 and type-2 paths with length l are simply given as $(\mathbf{A}^{(\tau)})^l$ and $(\mathbf{B}^{(\tau)})^l$. A simple extension to the path counting features in the single source case would be to use pure color paths only, i.e. type-1 and type-2. Counting the type-3 paths is more complicated since we want to distinguish two paths between two vertices not only by their lengths, but also by color of edges in the path. For example, we may want to weigh

$$\text{path 1: } i \xrightarrow{\mathbf{B}} i' \xrightarrow{\mathbf{A}} j' \xrightarrow{\mathbf{A}} j$$
$$\text{path 2: } i \xrightarrow{\mathbf{B}} i' \xrightarrow{\mathbf{B}} j' \xrightarrow{\mathbf{A}} j$$

differently, because edges of $\mathbf{A}$ could be more informative than edges of $\mathbf{B}$ in predicting the links in $\mathbf{A}^{(\tau+1)}$. In the supervised learning framework, we wish to have a feature for each particular color combination. Considering only undirected graphs, we require $g_{k,ij}(\cdot) = g_{k,ji}(\cdot)$, for any pair (i,j) and any k, and therefore count paths with reverse color patterns as the same. One can verify that the number of paths up to length $l_{\max}$ from all combination types are given by terms in the following matrix polynomial

$$(\mathbf{I} + \mathbf{A}^{(\tau)} + \mathbf{B}^{(\tau)})^{l_{max}}, \tag{15}$$

and, say, paths with pattern " $\circ \xrightarrow{\mathbf{B}} \circ \xrightarrow{\mathbf{B}} \circ \xrightarrow{\mathbf{A}} \circ$" can be counted efficiently using the matrix $\mathbf{B}^{(\tau)}\mathbf{B}^{(\tau)}\mathbf{A}^{(\tau)}$. With multiple auxiliary sources, denoted $\mathbf{B}, \mathbf{C}, \cdots$, the features in matrix form are given by

$$(\mathbf{I} + \mathbf{A}^{(\tau)} + \mathbf{B}^{(\tau)} + \mathbf{C}^{(\tau)} + \cdots)^{l_{max}}, \tag{16}$$

In practice, we may consider more than one retrospective step, and hence several separate multigraphs, each corresponding to a time step. To control the number of features, we do not allow any path combination between different time steps. Therefore in the case of k retrospective steps, the features set for predicting $\mathbf{A}^{(\tau)}$ are terms from

$$(\mathbf{I} + \mathbf{A}^{(\tau-k+1)} + \mathbf{B}^{(\tau-k+1)})^{l_{max}}, \cdots, (\mathbf{I} + \mathbf{A}^{(\tau)} + \mathbf{B}^{(\tau)})^{l_{max}}$$

4.4 Generalization to the Katz Measure

We now show that the score function (8) generalizes popular unsupervised models in several ways when using the path-counting features. From Section 4.3, when only considering the feature from the main network, the feature associated with length$-l$ paths is $\mathbf{G}_l(\mathbf{A}^{(t)}) = (\mathbf{A}^{(t)})^l$ in matrix form. The score function therefore becomes

$$\text{score} = \sum_{l=1}^{l_{max}} \theta_l^* \mathbf{G}_l(\mathbf{A}^{(t)}) = \sum_{l=1}^{l_{max}} \theta_l^* (\mathbf{A}^{(t)})^l. \tag{17}$$

Clearly (17) generalizes the truncated Katz measure by replacing the exponential damping factor β^l in (3) with a more general parameter θ_l, and hence introduces more modeling flexibility. With auxiliary sources like $\mathbf{B}^{(t)}$, the feature set will get much richer and the score function will have additional terms from $\mathbf{B}^{(t)}$, including

- powers of $\mathbf{B}^{(t)}$, corresponding to the pure color paths

- hybrid terms, such as $\mathbf{B}^{(t)}\mathbf{A}^{(t)}\mathbf{B}^{(t)}$, corresponding to the hybrid color paths.

Both types of terms, when properly weighted, could lend substantial prediction capability to the prediction model. Moreover, in practice, we may consider more than one retrospective step, which generalizes the truncated Katz even more.

4.5 Computational Complexity

A strict analysis of the computational complexity and a comparison between the logistic regression and least squares objectives is not straightforward. We present a discussion of the complexity and typical timings for one of our experimental settings. Assume we have a model that uses the adjacency matrices from t previous snapshots, m auxiliary sources, and highest number of path lengths l_{max}. Then the computational complexity, both for the logistic regression and least squares objectives, depends on the data matrix $\mathbf{S}$ that has $|\mathcal{N}_t \cup \mathcal{Z}_t| \approx N^2$ rows and $t(m+1)l_{max}$ columns. Here N is the number of vertices in an adjacency matrix and each column of $\mathbf{S}$ is related to a vectorization of an adjacency matrix or a power of an adjacency matrix. Since the matrices $\mathbf{A}^{(\tau)}$ and $\mathbf{B}^{(\tau)}$, $\tau = 1, \cdots, t$ are sparse, it follows that their powers $(\mathbf{A}^{(\tau)})^l$ and $(\mathbf{B}^{(\tau)})^l$ are sparse as well, but their density increases with l. The degree of densification of the power terms depends on the particular sparsity pattern of $\mathbf{A}^{(\tau)}$ and $\mathbf{B}^{(\tau)}$, which is different for different data sets. The efficiency of both logistic regression and least squares objectives is closely tied to the sparsity degree of $\mathbf{S}$. The least squares solution θ_{lsq} may be obtained using the associated normal equations yielding $\boldsymbol{\theta}_{\text{lsq}} = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{y}$, where $\mathbf{y}$ is the target vector in (12). The most expensive part in computing $\boldsymbol{\theta}_{\text{lsq}}$ is the product $\mathbf{S}^T\mathbf{S}$. Since the size of $\mathbf{S}^T\mathbf{S}$ is very small, i.e. $t(m+1)l_{max} \ll N^2$, the remaining calculations are negligible. Solving the logistic regression objective is an iterative process that usually involves computations of the gradient of the objective function. Of course the cost of computing the logistic regression gradients and the number of iterations to get a solution to a certain accuracy depends again on the sparsity degree and sparsity pattern of the data matrix $\mathbf{S}$. In Table 2 we present timings for the arXiv data set experiments using three different algorithms and for three different values of l_{max}.

5 Regularization

The path-counting features for multiple sources yield a rich set of features. Consider the case where we have c different sources, then the number of features associated with length-l paths is $(c^{l+1} - c)/(c-1)$, which is exponential in l. Thus the model fitting is prone to over-fitting as

1. the observations are extremely noisy, since the evolution of the social network is affected by many other unknown factors;

2. the dimensionality of feature space is high due to the multiple sources, with potentially many irrelevant or noisy features.

It is therefore important to control the complexity of the model through a proper regularization.

When predicting with multiple sources, it is often the case that some auxiliary sources do not contain information valuable for prediction. Also it is not hard to imagine that one particular path pattern (and therefore a feature) is not useful even though the component source in the feature is informative. These characteristics of our problem call for a sensible feature selection strategy, and it is therefore suitable to use recently proposed sparsity-promoting regularization schemes. As will be shown in our experiments (Section 6), the simple least squares objective is a rather effective surrogate of the logistic regression with much lower complexity. So, in this section we will focus only on the least squares objective.

Here we consider two strategies:

Lasso. We get the Lasso regression model ([9]) when we choose to put ℓ_1 regularization on parameters θ to filter out irrelevant features. With least squares fitting, the ℓ_1-regularized objective function is therefore

$$L_{lasso}(\boldsymbol{\theta}) = \|\mathbf{S}\boldsymbol{\theta} - \mathbf{y}\|^2 + \lambda\|\boldsymbol{\theta}\|_1, \tag{18}$$

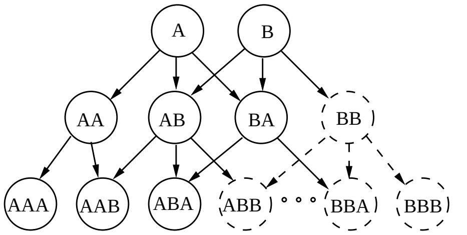

Figure 3: An illustration of the hierarchical sparsity from multiple sources.

where λ controls the sparsity of the learned parameter $\boldsymbol{\theta}$. This regularization can be applied to both pure color paths and hybrid paths.

Hierarchical Sparsity. The ℓ_1 regularization in Lasso is flat in the sense that it puts uniform regularization on all features. More sophisticated group-Lasso based regularization designs consider the hierarchical structure inherent to the task [2, 28]. The structure in our problem lies in the way the composite paths are constructed. Basically, we wish that if a path pattern ("feature") ω is knocked out, all the path patterns containing ω as sub-pattern should receive zero weight too. For example, if particular path pattern "$\circ \xrightarrow{\mathbf{B}} \circ \xrightarrow{\mathbf{A}} \circ$" (or equivalently the feature in matrix form $\mathbf{BA}$) has zero weight, we wish that all the features which contain $\mathbf{BA}$, e.g., $\mathbf{BAA}$, $\mathbf{B}^2\mathbf{A}$ or $\mathbf{BA}^2\mathbf{B}$, to be excluded from the feature set. This relation between features can be fully expressed as a *directed acyclic graph* (DAG), as illustrated in Figure 3, where the arrow (directed edge) between vertices (path patterns) shows the "containing" relation between two path patterns.

To enforce this kind of feature preference, we can use the composite absolute norm introduced by Zhao et. al [28]. This is implemented through group Lasso with overlapping groups. With a DAG $(\mathcal{V}, \mathcal{E})$, a group $\mathcal{G}_v \subset \mathcal{V}$ with "root" node v contains v and all of its offsprings, and the set of all such groups is therefore

$$\mathcal{G} = \{v \cup \text{all offsprings of } v | v \in \mathcal{V}\}.$$

The ℓ_∞ norm for each group $g \in \mathcal{G}$ is defined as

$$\|\boldsymbol{\theta}_g\|_\infty = \max_{v \in g} |\boldsymbol{\theta}_v|$$

and the composite norm is simply

$$\|\boldsymbol{\theta}_\mathcal{G}\|_c = \sum_{g \in \mathcal{G}} \|\boldsymbol{\theta}_g\|_\infty.$$

The overall cost function with this structured sparsity penalty is hence defined as

$$L_{structure}(\boldsymbol{\theta}) = \|\mathbf{S}\boldsymbol{\theta} - \mathbf{y}\|^2 + \lambda \|\boldsymbol{\theta}_\mathcal{G}\|_c. \tag{19}$$

In Figure 3 we show two sources that are mixed together to form a DAG up to power $l = 3$. Suppose the feature $\mathcal{G}_{\mathbf{BB}}$ (denoted by dashed circles) is filtered out. Then the set of selected variables will be

$$\mathbf{A},\ \mathbf{B},\ \mathbf{AA},\ \mathbf{AB},\ \mathbf{AAA},\ \mathbf{AAB},\ \mathbf{ABA},\ \mathbf{BAB}$$

and comply with the sparsity structure as well as the hierarchy of the DAG. Note also this sparsity structure is promoted but not enforced through regularization (group Lasso), and in practice the "undesired" feature combination could still appear, especially when the regularization parameter is not large enough.

6 Experimental Results

We have conducted extensive tests of our prediction model on a variety of collaboration networks. We use the Katz and truncated Katz as the representatives of unsupervised models, because of their overall good performance in link prediction [19] and their close relation to our path-counting features. Within the proposed supervised framework, we also intend to compare ones with single source and multiple sources, as well as models with different feature designs and regularization.

Table 1: Some statistics of arXiv (HepTh), CiteSeer (CS-1, CS-2, CS-3) and SIAM data sets. Here ‘core” denotes the number of authors who have published at least one paper in any given snapshot during training, “train” denotes number of links among authors that appear in the training set, and “test” denotes the number of links among authors in the testing set.

data set		HepTh	CS-1	CS-2	CS-3	SIAM
core		1381	2321	2448	1182	6891
train	**A**	14507	9683	13634	8703	5528
	B	61674	35509	41956	21781	5431
	C	22075	11269	16294	11988	5124
	D	39596	19776	30931	20283	6504
test	**A**	6788	6809	9609	6262	2764
	B	20523	17067	19464	12764	2715
	C	15459	6591	10100	6297	2562
	D	11576	12643	18799	9860	2586

6.1 The Data Sets

We adopted three real-world data sets: arXiv-HepTh, CiteSeer and SIAM from the scientific publication domain, and constructed evolving social and proximity networks between authors based on their publication history.

- arXiv-HepTh: publications from 1992 to 2003 in high energy physics-theory (hep-th) section from e-Prints at arXiv (www.arxiv.org). We formed four snapshots, each from the publications from non-overlapping intervals of three years;

- CiteSeer: publications in the Scientific Literature Digital Library from 1995 to 2003. Since in this data set there are very few number of authors who continuously published papers every year, we further divided it into three subsets to get three relatively large data sets: CiteSeer-1 contains publications from 1995 to 1997; CiteSeer-2 contains publications from 1998 to 2000; and CiteSeer-3 contains publications from 2001 to 2003. For each subset, we formed three snapshots, each based on the publications of one year.

- SIAM: publications in 11 journals and proceedings for the period 1999-2004 hosted by the Society of Industrial and Applied Mathematics (SIAM). Unfortunately we do not have time stamps for the publications. On this data set, we need to artificially generate a missing link prediction problem (see Section 6.2 for more detail).

Various relationships and proximity measures between authors have been extracted from their publications as summarized below, and therefore several networks are formed with the authors being the vertices.

- **A** (co-authorship): $A_{ij} = 1$ iff author i and author j co-authored at least one paper;
- **B** (co-citation): $B_{ij} = 1$ iff author i and author j have cited same papers;
- **C** (co-reference): $C_{ij} = 1$ iff papers by author i and author j are cited by the same paper;
- **D** (text similarity): $D_{ij} = 1$ iff the cosine similarity between papers (represented with the “bag-of-words“ model) published by author i and author j is over a threshold.

Although networks **A**, **B** and **C** all indicate social relations with varying levels, we will predict on co-authorship **A** with other networks **B**, **C**, **D** as auxiliary data. Results using networks **B** and **C** as targets have also been obtained and are similar in spirit to the results presented here (and will be made available in the technical report.) Some statistics of each data set are given in Table 1.

6.2 Experimental Settings

For any data set with $t+1$ snapshots, we use snapshot 1 to t for training, and $t+1$ for testing. This applies to arXiv-HepTh and CiteSeer, but on SIAM we do not have time stamps for the links in these networks. In this case, we transform the problem into missing link discovery, as suggested in [17]. The procedure is to randomly split the author-author links in $\mathbf{A}$ into three parts: the training (44% of links, denoted $\mathbf{A}^{(1)}$), the validation (22% of links, denoted $\mathbf{A}^{(2)}$) and the test part (33% of links, denoted $\mathbf{A}^{(3)}$). In the training phase we learn the model parameter $\boldsymbol{\theta}^*$ through predicting $\mathbf{A}^{(2)}$ using $\mathbf{A}^{(1)}$ and auxiliary networks $\mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ for the corresponding snapshots. In testing, we apply the model $\boldsymbol{\theta}^*$ to a combined training set $\mathbf{A}' = \mathbf{A}^{(1)} + \mathbf{A}^{(2)}$.

Once the model parameter $\boldsymbol{\theta}$ is learned, we then apply to the snapshot from different sources at time $t+1$ to get the scores. A pair is predicted to have link if its score is over a certain threshold h. Clearly a smaller threshold gives a more "aggressive" predictor which predicts more pairs to be links.

The performance of models are evaluated in two different yet related ways:

- *Precision:* we select $|\mathcal{N}_t|$ "feasible" pairs with highest scores as our predictions of new links for time $t+1$ and calculated the proportion of true links in terms of percentage.
- *ROC Curve*: a receiver operating characteristic (ROC) curve graphically represents the true positive rate *vs.* false positive rate as the threshold for prediction changes. For our problem, the true positive rate and false positive rate are calculated respectively as

$$\begin{aligned} r_{\text{true}} &= \frac{\text{number of correctly predicted links}}{\text{number of true new links in } \mathbf{A}^{(t+1)}} \\ r_{\text{false}} &= \frac{\text{number of incorrectly predicted links}}{\text{number of non-linked pairs in } \mathbf{A}^{(t+1)}}. \end{aligned}$$

  We are particularly interested in the range of the ROC curve with small false positive rate, which corresponds to large threshold on scores and hence less aggressive link prediction models. This prediction setting better mimics the real world scenarios, such as friends recommendation in an on-line social network.

6.3 Least-squares vs. Logistic Regression

We have discussed in Section 4.2 the square loss as a cheap surrogate for the logistic regression objective in training the prediction models. Here we give empirical comparison of logistic regression and its square loss surrogate with arXiv-HepTh as the representative data set. More specifically, we compare the following three prediction models:

- SL-SINGLE: supervised learning with single source;
- SL-PURE: supervised learning using all pure color paths;
- SL-HYBRID: supervised learning using hybrid color paths.

The result in terms of precision and time complexity is given in Table 2. As it shows, least square objective yields performance comparable to more expensive logistic regression using features with different levels of richness. In the remainder of the experiments, we will use the square loss as the surrogate for logistic regression objective.

6.4 Models to Evaluate

We will give a detailed exposition of results from various models discussed in Section 4 and Section 5. In addition to the unregularized supervised models discussed in Section 6.3, we will also test on unsupervised models and supervised models with sparsity-promoting regularization:

Table 2: Comparison between least-squares and logistic regression in terms of precision (in percentage) and time complexity (in seconds) on arXiv-HepTh data set.

		least squares		logistic regression	
	$l_{\max}$	precision	time (sec.)	precision	time (sec.)
SL-single	2	2.30	1.13	1.95	43.18
	3	2.12	1.38	2.12	49.52
	4	2.83	1.56	2.48	55.67
SL-pure	2	1.95	1.92	2.12	75.92
	3	1.77	3.23	1.24	96.34
	4	2.12	4.75	0.88	115.77
SL-hybrid	2	1.95	1.64	2.83	35.20
	3	2.12	17.63	1.77	182.41
	4	2.48	358.99	1.59	3098.68

Unsupervised Models

- Katz-single: Katz measure based on single source ($\mathbf{A}$) with optimal parameter β;
- Katz-combined: Katz measure based on combined adjacency matrix from multiple sources ($\mathbf{F} = \mathbf{A} + \mathbf{B} + \mathbf{C} + \mathbf{D}$) with optimal parameter β;
- tKatz-single: truncated Katz based on single source ($\mathbf{A}$) with optimal parameter β;
- tKatz-combined: truncated Katz based on combined adjacency matrix with optimal parameter β;

Supervised Models with Regularization

- SL-pure(l): supervised learning using all pure color paths with square loss and ℓ_1 regularization;
- SL-hybrid(l): supervised learning with hybrid color path features with square loss and ℓ_1 regularization;
- SL-hybrid(g): supervised learning with hybrid color path features with square loss and regularization promoting hierarchical sparsity structure (implemented in group Lasso, see Section 5).

We intend to study the impact of supervision signal, features, and regularization to the prediction accuracy. This is shown through empirical comparison on the following specific sets of models:

- supervised models (SL-single, SL-pure, SL-hybrid and their regularized versions) *Vs.* unsupervised models (Katz-single, Katz-combined, tKatz-single, tKatz-combined);
- single source models (Katz-single, tKatz-single, SL-single) *Vs.* multiple source ones (Katz-c, tKatz-c, SL-pure, SL-hybrid)
- supervised models with simple features (SL-single, SL-pure) *Vs.* models with rich features (SL-hybrid)

6.5 Results and Analysis

The performance of the above mentioned models are reported in Table 3 and Figure 4. Table 3 contains the precision results on all the data sets given by all the models. Figure 4 further compares ROC curves for a number of selected supervised models in the interesting region of small false positive rate ($\leq 0.2\%$). A simple comparison shows that the ROCs and precision numbers tell a similar story. Basically we achieve improved performance with supervised models and multiple sources.

More specifically, as discussed in Section 6.4, we compare the performance of models from three different perspectives.

Analysis I: The Role of Supervision. The clear message from Table 3 and Figure 4 is "supervision helps in link prediction", which holds in the following two senses:

- The single-source supervised model SL-single has the *same* path-counting features based source $\mathbf{A}$ as the unsupervised Katz-single and tKatz-single, while SL-single is overall better than its unsupervised counterparts. This is not surprising, since the unsupervised model has only one parameter

Table 3: Link prediction results in terms of precision on three CiteSeer subsets, arXiv-HepTh data set and SIAM data set.

	$l_{\max}$	CiteSeer-1	CiteSeer-2	CiteSeer-3	arXiv-HepTh	SIAM
Katz-single		19.3	13.2	20.4	2.26	32.6
Katz-combined		21.1	14.8	15.3	0.38	34.6
tKatz-single	2	19.5	16.4	21.6	1.41	41.1
	4	19.3	13.2	20.4	2.26	32.2
tKatz-combined	2	21.1	14.8	15.0	0.38	32.1
	4	21.1	14.8	15.0	0.38	32.0
SL-single	2	19.6	16.8	19.2	2.30	41.5
	4	24.1	21.2	24.6	**2.83**	41.5
SL-pure	2	19.3	16.1	15.1	1.95	49.2
	4	24.1	23.4	22.5	2.12	50.7
SL-pure(l)	2	19.3	16.1	15.1	2.08	49.2
	4	24.1	23.4	22.5	2.21	50.6
SL-hybrid	2	32.5	27.3	34.2	2.08	50.9
	4	32.6	27.3	34.3	2.48	52.3
SL-hybrid(l)	2	32.5	27.3	34.0	2.48	50.8
	4	32.5	27.3	34.0	2.12	52.3
SL-hybrid(g)	2	**33.9**	27.7	**34.9**	1.95	51.1
	4	33.4	**27.9**	34.2	2.48	**52.6**

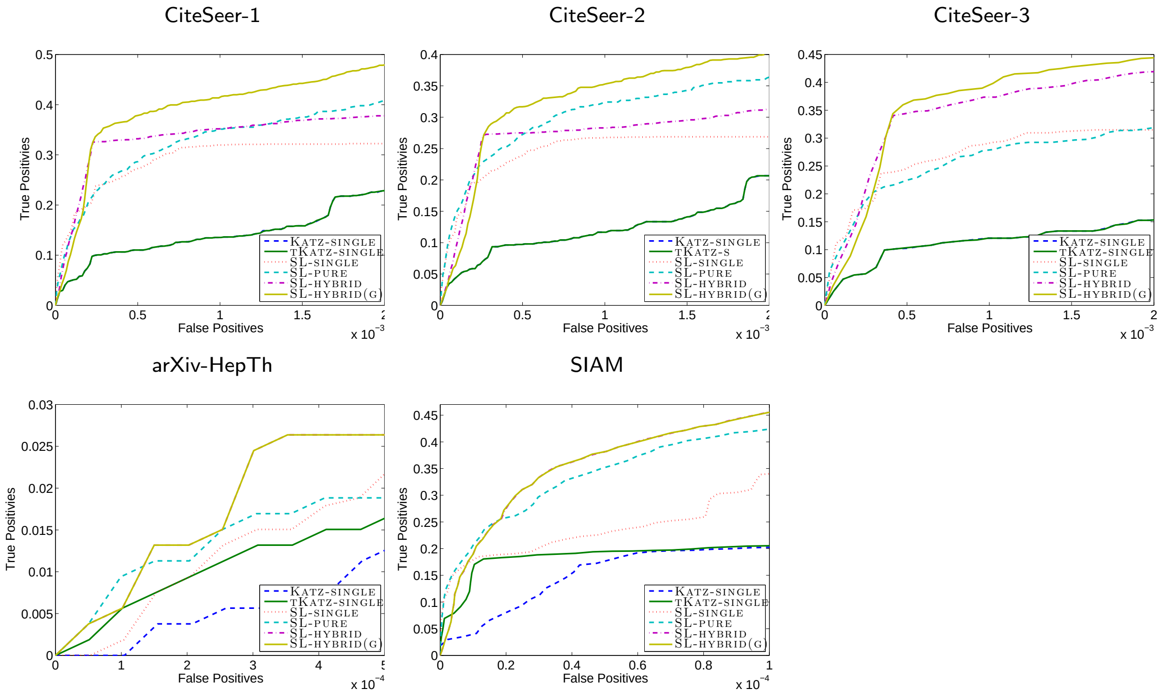

Figure 4: The ROC curves for the selected unsupervised and supervised learning models.

β, while SL-single has more parameters for different powers (path lengths) and retrospective steps, which can be learned effectively through the supervised framework.

- Supervision helps in learning a proper way to synthesize information from multiple auxiliary sources, as in models SL-PURE, SL-HYBRID and their regularized versions. This turns out to be much more effective than the naive way to combine different sources, as adopted by unsupervised KATZ-COMBINED and TKATZ-COMBINED.

Analysis II: The Role of Multiple Sources. First of all, multiple auxiliary sources greatly help the prediction on the target network in the supervised framework. On all data sets except arXiv-HepTh, multiple-source supervised models (SL-PURE, SL-HYBRID, and their regularized versions) are clearly better than the single-source supervised models SL-SINGLE, especially when the information from auxiliary sources are encoded in a richer feature set. Secondly, the auxiliary sources could also be distractive and irrelevant, and therefore hurt the performance of the predictor when inappropriately integrated into the system. This aspect is most clear from the comparison of single-source unsupervised models (KATZ-SINGLE, TKATZ-SINGLE) and their multiple-source counterparts (KATZ-COMBINED, TKATZ-COMBINED). From Table 3, it is not rare that a naive combination of network sources in KATZ-COMBINED and TKATZ-COMBINED yields inferior performance than the single source unsupervised model. arXiv-HepTh is an interesting example, on which we observe over 80% decrease of accuracy when using multiple sources in an unsupervised way. However, we argue that one attractiveness of our framework is that the supervised framework can effectively discriminate useful auxiliary sources and features from the irrelevant and distractive ones, therefore minimizing the harm. Indeed, on arXiv-HepTh where the auxiliary sources are harmful when used in a naive way, the multiple-source supervised models SL-PURE, SL-HYBRID, and SL-HYBRID(G) still manage to give good performance.

Analysis III: Feature Design and Regularization. We are clearly benefiting from the rich set of features. It can be seen by comparing the two multiple-source models SL-PURE, the model with pure color paths as features, and SL-HYBRID, the model with hybrid color paths as features. For all prediction tasks, SL-HYBRID performs better than SL-PURE, showing the predictive power of cross-source paths in feature design. Moreover, the regularization promoting structured sparsity helps to further improve the accuracy. Indeed, for all the tasks SL-HYBRID(G) is better than SL-HYBRID and SL-HYBRID(L).

7 Conclusion and Discussion

In this paper, we have proposed a novel and general framework of supervised link prediction using multiple sources of data. Different from the commonly used unsupervised link prediction methods, our model can effectively and efficiently learn the network dynamics from a time series of network snapshots, and therefore improve the link prediction accuracy. In addition, multiple graphs over the same set of nodes but from different sources can be naturally incorporated into the supervised framework. We have performed extensive set of experiments on three real-world data sets. The experimental results confirm that prediction accuracy can be improved both using supervision and multiple sources of information.

Despite the empirical success of the proposed model, a few directions remain to be explored. First, we haven't fully exploited our models' ability on taking features other than path counts. As suggested in [10,18], a lot of microscopic features and other network local/global characteristics can be informative for link formation, most of which can be readily used in our supervised framework. Second, it is still unclear what probabilistic model is most appropriate for the predictive modeling of links. For example, we could alternatively adopt the Prackett-Luce ranking model [27] to describe the latent mechanism of link generation, and view all the new links as observed to be top-ranked. Third, in real-world applications, the independence assumption (3.1) might be considered to be too restrictive. One choice to relax it is to partition the network into many small super-nodes (node clusters), and assume independence between super-nodes, much like the relaxation to mean field approximation in variational methods [11]. Finally, many social networks are massive in size and therefore pose a scalability issue [25]. In future work, we plan to address and conduct research on all these issues.

References

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25:211–230, 2003.

[2] F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, 2008.

[3] A. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311:590–614, 2002.

[4] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. In *PNAS*, volume 106, 2009.

[5] F. Guo, S. Hanneke, W. Fu, and E. Xing. Recovering temporally rewiring networks: a model-based approach. In *ICML'07*, 2007.

[6] S. Hanneke, W. Fu, and E. Xing. Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4(2010) 585–605, 2010.

[7] F. Harary. *Graph Theory.* Westview Press, 1994.

[8] M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Workshop on Link Analysis, Counterterrorism and Security (SDM)*, 2006.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer-Verlag, New York, 2001.

[10] Z. Huang. Link prediction based on graph topology: The predictive value of the generalized clustering coefficient. In *Workshop on Link Analysis (KDD)*, 2006.

[11] T. Jaakkola. Tutorial on variational approximation methods. In *In Advanced Mean Field Methods: Theory and Practice*, 2000.

[12] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms, arXiv:0904.3523, 2009.

[13] H. Kashima and N. Abe. A parameterized probabilistic model of network evolution for supervised link prediction. In *ICDM'06*, 2006.

[14] H. Kashima, T. Kato, Y. Yamanishi, M. Sugiyama, and K. Tsuda. Link propagation: A fast semi-supervised learning algorithm for link prediction. In *SDM'09*, 2009.

[15] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

[16] L. Katz and C. Proctor. The concept of configuration of interpersonal relations in a group as a time-dependent stochastic process. *Psychometrika*, 24:317–327, 1959.

[17] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML'09*, 2009.

[18] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD'08*, 2008.

[19] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM'03*, 2003.

[20] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251, 2004.

[21] M. Newman. Clustering and preferential attachment in growing networks. *Phys. Rev. E*, 64(025102), 2001.

[22] G. Robins and P. Pattison. Random graph models for temporal processes in social networks. *Journal of Mathematical Sociology*, 25:5–41, 2001.

[23] T. Snijders. Statistical methods for network dynamics. In S. L. et al., editor, *Proc. of the XLIII Scientific Meeting, Italian Statistical Society*, pages 281–296, 2006.

[24] T. Snijders, G. van de Bunt, and C. Steglich. Introduction to stochastic actor-based models for network dynamics. *Social networks*, 2009.

[25] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu. Scalable proximity estimation and link prediction in online social networks. In *IMC'09*, 2009.

[26] B. Taskar, M.-F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In S. Thrun, L. Saul, and B. Schölkopf, editors, *NIPS 16*, Cambridge, MA, 2004.

[27] J. Guiver and E. Snelson. Bayesian inference for Plackett-Luce ranking models. In *ICML'09*, 2009.

[28] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, 37:3468–3497, 2009.