Towards Time-aware Link Prediction in Evolving Social Networks

Tomasz Tylenda, Ralitsa Angelova, Srikanta Bedathur
Max-Planck Institute for Informatics
Saarbrücken, Germany
{ttylenda, angelova, bedathur}@mpi-inf.mpg.de

ABSTRACT

Prediction of links - both new as well as recurring - in a social network representing interactions between individuals is an important problem. In the recent years, there is significant interest in methods that use only the graph structure to make predictions. However, most of them consider a single snapshot of the network as the input, neglecting an important aspect of these social networks viz., *their evolution over time*.

In this work, we investigate the value of incorporating the history information available on the interactions (or links) of the current social network state. Our results unequivocally show that time-stamps of past interactions significantly improve the prediction accuracy of new and recurrent links over rather sophisticated methods proposed recently. Furthermore, we introduce a novel testing method which reflects the application of link prediction better than previous approaches.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; G.2.2 [**Graph Theory**]: Network Problems

Keywords

Social Networks, Evolution, Link Analysis, Link Prediction

General Terms

Algorithms

1. INTRODUCTION

Social networks are ubiquitous – ranging from academic collaboration networks to terrorist networks, and from telephone calls between subscribers to online social interaction via Facebook, Orkut, LinkedIn, etc. These networks can be represented as graph and hypergraph structures where nodes represent people or other entities involved in the social interaction and an edge/hyperedge between them indicates presence of an interaction. Accurate prediction of new link formation or repeated occurrences of existing links is an important problem in this setting. Such *link prediction* is useful in a variety of application scenarios: to predict the next academic collaborations between researchers, detection of unseen links in terrorist networks, and for suggesting new friendships in online social networking sites.

The link prediction problem has attracted immense interest in recent years, and a variety of techniques that operate on the graph/hypergraph structure of social networks are proposed. All link prediction methods address the following question: *"Given a pair of nodes u and v in the current social network, how likely is it that u will interact with v in the future?"*.

However, current approaches based on answering the above question have two important drawbacks. First of all, barring a few [18, 17], none of the proposed methods take into account the *evolutionary history* of social networks for link prediction tasks. Networks continuously evolve in response to the underlying social dynamics. Clearly, older events are less likely to be relevant for determining the future linkages than recent ones. Second, in many social network settings, it is useful to provide for a chosen node u, a ranking of candidate nodes based on their probability of future interaction with the chosen node. However, current methods of link prediction focus on estimating the probability of having a link between a specified u and v.

In this work, we study the impact of considering the temporal evolution of social networks explicitly in link prediction tasks, and make following contributions:

1. We develop graph-based link prediction techniques that incorporate the temporal information contained in evolving social networks. In particular, we extend the local probabilistic model proposed by Wang et al. in [19] to include *time-awareness*. We show how to incorporate edge weights – possibly derived from temporal features– into the state-of-the-art link prediction methods, such as the Adamic-Adar distance based and rooted PageRank based techniques [11].

2. We introduce a novel testing method that evaluates the performance of different prediction algorithms in their ability to *rank nodes from a neighborhood of a selected node*. We propose two measures of the quality of the rankings, which focus on different aspects of rankings.

We conducted an empirical evaluation of our techniques over two collaboration networks commonly used in the earlier link-prediction research – DBLP [6] and `astro-ph` slice of ArXiv.

The 3rd SNA-KDD Workshop '09 (SNA-KDD'09), June 28, 2009, Paris, France. Copyright 2009 ACM 978-1-59593-848-0...$5.00.

Our experiments show that incorporating time-based weights significantly improves the prediction performance of all the methods.

The rest of the paper is organized as follows: Section 2 presents the data model and notations we use, and briefly provides a background on link prediction tasks studied in the literature. Next, Section 3 outlines various time-agnostic, state-of-the-art link prediction methods. Section 4 which forms a key contribution of this paper, describes in detail how to incorporate temporal features of an evolving social network for link prediction. We introduce our novel evaluation methodology in Section 5 and experimental results in Section 6. Finally, we briefly sketch the related work in Section 7 before concluding in Section 8.

2. BACKGROUND

2.1 Data Model and Notations

Social network data can be naturally represented as a hypergraph $H = (V, E)$. Nodes represent users (authors) and hyperedges represent social interaction between them (co-authorship of a paper). The hypergraph H can be transformed into a simple graph $G = (V, E')$, where two nodes are connected iff there exists a hyperedge $e \in E$ connecting them. Although the simple graph representation of a network contains less information than a hypergraph, link prediction methods over the simple graph can benefit from reduced noise in the data. In the rest of the paper, we use *author* and *node* interchangeably since we deal only with bibliographic networks in this work.

2.2 Tasks in Link Prediction

O'Madadhain et al. pointed out [16] that social networks can represent two kinds of data: (i) *persistent relations*, e.g. friendships between two people and (ii) *discrete events*, e.g. co-authorship of scientific papers. In the first situation the network is modeled by a simple graph and new edges are created between vertices that are at distance two or higher. In the second situation, however, edges may appear not only between vertices at distance two or higher but also between vertices that are already connected (repeated edges). As we show later, the number of repeated edges in co-authorship networks can be as high as 50%. We can reasonably expect similar, if not larger, fraction of repeated edges in other networks corresponding to discrete events, such as telephone call networks or e-mail networks.

The above observations suggest that there are two main problems in link prediction: (i) prediction of *new links*, which we also call as *prediction at distance 1*, (ii) prediction of *both new and repeated links*, which we call *prediction at distances 2 and higher*. Current link prediction methods are mostly tested in the first scenario. Surprisingly, little effort was made to solve the second problem. Although one can expect that methods of predicting new links also work in the case of repeated links, we found that exploiting temporal information can be particularly beneficial for repeated link prediction.

2.3 Edge-centric vs. Node-centric Prediction

The link prediction problem we stated earlier, requires as input two nodes v and u between which the probability of an edge is estimated. We term this as an edge-centric approach to link prediction. To the best of our knowledge, all publications about link prediction follow the edge-centric approach: the link prediction problem is stated as a ranking of vertex pairs. The edge-centric approach has its disadvantages. It assumes that users are interested in new edges irrespective of the vertices they connect. This may lead to situations where the top of the list of edges is occupied by mutually uninteresting pairs of vertices. In our opinion, users are typically interested in the growth of the network around a selected vertex. Given v, a user wants to know which vertices are most likely to be connected with v in the future. This is related to tasks of discovering new and old friends in social networks, e.g. the *People You May Know* tool on the social networking website Facebook.

Edge-centric link prediction often boils down to deciding whether two nodes have anything in common – irrespective of the distance between them in the social network. However, in reality, most links are established between nodes that are close in terms of the graph distance between them and similarity of attributes. These facts are known as the *locality of edge attachment* [9] and *homophily* [10, 15]. When the link prediction is treated as a binary classification, the following problem occurs in the selection of training and testing data points. The negative points are random pairs of vertices. Therefore, it is highly unlikely that they have any co-authors in common, that they work in the same field, or even that they are connected by some path. It would make more sense to choose pairs of vertices that are close to each other, but did not collaborate.

In order to avoid this problem, we propose to utilize a node-centric approach. Given a node v we calculate its neighborhood $\mathcal{N}(v)$ and output a list of vertices $w \in \mathcal{N}(v)$ according to the propensity that they will collaborate with v. The neighborhood $\mathcal{N}(v)$ does not be a neighborhood as defined in graph theory, i.e. elements of $\mathcal{N}(v)$ need not be linked with v. It makes sense, however, to build the neighborhood based on graph-theoretic measures. For instance, we can define $\mathcal{N}(v)$ as the direct and the distance-2 neighbors of v. This is a more natural view of the link prediction problem than the edge-centric approach. Moreover, we consider only pairs of vertices that are close to each other. Since some of the nodes collaborated with the central node recently and others a long time ago, link prediction methods benefit from time-awareness in the node-centric scenario. This is not the case in the edge-centric approach, which focuses on deciding whether nodes are related in any way.

3. TIME-AGNOSTIC LINK PREDICTION

We start the presentation of link prediction methods with a brief description of time-agnostic methods that we will extend to be time-aware. All link prediction methods that we consider are functions $f : V \times V \to \mathbb{R}$, where one of the arguments is the central node. The value of the function is used to rank the nodes from the neighborhood of the central node.

3.1 Baseline Methods

The baseline methods we use are described in detail by Liben-Nowell and Kleinberg [11]. Let $\Gamma(x)$ denote neighbors of the node x, we define the following measures

$$|\Gamma(x) \cap \Gamma(y)| \qquad \text{common neighbors} \tag{1}$$

$$\frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \qquad \text{Jaccard's coefficient} \tag{2}$$

A robust measure similar to common neighbors was proposed by Adamic and Adar [1]. Closeness between nodes x and y is defined as:

$$AA\,(x, y) := \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma\,(z)|} \tag{3}$$

The intuition behind this method is simple. It measures the number of common neighbors of x and y. Each neighbor z is weighted according to its importance for the connection. If a node has many neighbors, the fact that it connects x and y is not very meaningful. On the contrary, if it has only a few neighbors, then it is important for the connection.

The rooted PageRank measure is more subtle and achieves good results. It is the stationary probability that a random walk from x visits node y. To minimize the influence of distant parts of the graph on the value of the predictor, the walk jumps to the starting node with probability α. For efficiency reasons we calculate PageRank only on the neighborhood of the central node.

3.2 Time-agnostic Maximum Entropy

In this section, we will present details of local probabilistic model for link prediction proposed by Wang et al. [19]. Since the method is based on the maximum entropy principle we will refer to it as the time-agnostic maximum entropy method. Let us consider an example:

$$V = \{a, b, c, d, e, f\} \qquad \mathcal{N} = \{a, b, c, d\}$$

If a and d wrote a paper together, then the corresponding event x would be $\langle 1, 0, 0, 1\rangle$. If the paper was written by a, d and e, the string would be the same since $e \notin \mathcal{N}$. The empty set – $\langle 0, 0, 0, 0\rangle$ – denotes any paper written by authors who are not in $\mathcal{N}$. Usually $|\mathcal{N}|$ is much less than $|V|$, hence we expect that $P(\emptyset)$ will be close to 1.

Let u and v be authors, the probability that they collaborate is a marginal of P:

$$P(u, v) = \sum_{\{x \in S | \{u,v\} \subseteq x\}} P(x). \tag{4}$$

Our main task is to estimate the joint probability distribution. The hypergraph describing the co-authorship network contains events (papers) that we observed. Compared to the number of possible events (2^k), our data set is very sparse. Therefore, P must be estimated carefully – we want the estimator of P to be "smooth". This goal can be achieved by using the *maximum entropy principle* [8]. The principle says that we should choose the distribution which: (i) agrees with the observations we have about P, (ii) does not make any additional assumptions. *No additional assumptions* means that P should be as close as possible to a prior distribution Q. The distance is measured by Kullback-Leibler divergence:

$$D(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

In the special case of $Q = U$ (uniform distribution), minimizing the divergence is equivalent to maximizing entropy of distribution P, hence the name of the method — *maximum entropy estimation*. In our setting, we use a prior different from the uniform as follows: Let $P(a_i)$ be the probability that author a_i writes a paper. Then the prior probability of a paper is

$$Q\left((a_1, a_2, \ldots, a_n)\right) = \prod_{i:a_i=1} P(a_i) \cdot \prod_{i:a_i=0} (1 - P(a_i))$$

This prior is equivalent to assumption that an author of a paper occurs independently of other authors.

What does P *agrees with observations* mean? Suppose that our set of observations contains 200 elements (papers) and exactly 10 of them are co-authored by u, v and perhaps by some additional authors. We impose a constraint on the joint distribution. It is represented by the marginal $P(u, v) = 1/20$. The constraint can be also written as

$$\sum_{x \in S} P(x) k(x) = \frac{1}{20}$$

where k is an indicator function

$$k(x) = \begin{cases} 1 & \text{if } \{u, v\} \subseteq x \\ 0 & \text{otherwise} \end{cases}$$

Finally, the problem of finding the maximum entropy distribution can be stated as

$$\begin{aligned} \min_P \quad & D(P \parallel Q) \\ \text{s.t.} \quad & \sum_{x \in S} P(x) k(x|j) = d(j) \quad \text{for } j = 0, \ldots, m \end{aligned} \tag{5}$$

3.2.1 Lagrange Multipliers for Time Agnostic Problem

Lagrange multipliers is a method of optimizing a function subject to some constraints. For a detailed description of the method we refer the reader to [4]. Applied to the problem formulated in (5) it results in:

$$\begin{aligned} L(P, \lambda) = & \sum_{x \in S} P(x) \log \frac{P(x)}{Q(x)} \\ & + \sum_{j=0}^{m} \lambda_j \left(\sum_{x \in S} P(x) k(x|j) - d(j) \right) \end{aligned} \tag{6}$$

We want to obtain the dual function

$$g(\lambda) = \inf_P L(P, \lambda) \tag{7}$$

We do this by minimizing over P by setting the appropriate derivatives to 0:

$$\frac{\partial L}{\partial P(x)} = \log \frac{P(x)}{Q(x)} + 1 + \sum_{j=0}^{m} \lambda_j k(x|j) \overset{!}{=} 0$$

we obtain

$$P(x) = Q(x) \exp\left(-1 - \sum_{j=0}^{m} \lambda_j k(x|j) \right) \tag{8}$$

This can be substituted into (7). In order to make the formulas "match" the time aware version that follows, we denote λ_0 by ν:

$$g(\lambda, \nu) = \sum_x Q(x) \exp\left(-1 - \sum_{j=1}^{m} \lambda_j k(x|j) - \nu \right) + \sum_{j=1}^{m} \lambda_j d_j + \nu \tag{9}$$

Under the strong duality the solution to the dual problem gives us also the solution to the primal problem. Therefore, we need to solve just the dual. In the time aware version of maximum entropy, described below, we require the solution of time agnostic problem. We refer the value of λ which optimizes (9) as l.

4. TIME-AWARE LINK PREDICTION

This section presents our time-aware link prediction methods. We we start we a presentation of extensions to baseline methods, in the order of maximum distance they can handle. Then, we move to the time-aware maximum entropy.

4.1 Extensions to Baseline Methods

4.1.1 Repeated Link Prediction

The first task that we consider is repeated link prediction. Suppose, we want to predict the next repeated collaboration of node v. We treat the problem as a ranking of nodes from $\mathcal{N}(v) = \{w \mid w \neq v \wedge distance(v, w) = 1\}$. The ranking of node w is based solely on the properties of the edge $\langle v, w \rangle$. In the co-authorship network there are three properties that can be used for this purpose:

Time – nodes are sorted according to the time that elapsed since their last collaboration with the central node.

Number of collaboration events – the sorting criterion is the number of collaborations with the central node.

Minimal number of co-authors – if a paper is written by two or three authors, they probably collaborate closely. On the contrary, if there are 15 authors, not all the pairs worked closely together. Thus, we sort $w \in \mathcal{N}(v)$, by the minimal number of authors over papers that connect v and w:

$$\min_{\{e \in E \mid v, w \in e\}} degree(e),$$

where $degree(e)$ is the number of hypervertices connected by a hyperedge e.

4.1.2 Prediction at Distance 1 and 2

In prediction at distance 1 and 2 we used an extended version of the Adamic-Adar method (3):

$$AA(x, y) := \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w(x, z) \cdot w(z, y)}{\log |\Gamma(z)|}.$$

The weighting function $w(\cdot, \cdot)$ should reflect the strength of connection between two vertices. We use the same three criteria that we used for link prediction at distance 1:

- **Age of the most recent collaboration:** Weighting function is the same as the weighting in the maximum entropy methods. Concrete functions are presented in Section 6.
- **Number of collaborations:** $w_c(u, v) := \log_2(|\Lambda(u) \cap \Lambda(v)| + 1)$
- **Minimal number of co-authors:** $w_s(u, v) := 1/\log_2(\text{minimal number of co-authors})$

4.1.3 Prediction at Any Distance

We used two methods that can predict links between nodes at any distance, i.e. they do not become zero as the distance between nodes reaches some threshold.

Rooted PageRank. The rooted PageRank can be naturally extended with edge weighting. In the simplest case the probability of following a link of the current node is uniform. We can introduce time-awareness, as well as other extensions, by following a link with probability proportional to its weight. Let us denote the weight of an edge by $w(\cdot, \cdot)$. If x and y are neighbors, then the probability of the walk going from x to y is

$$P[x \to y] = (1-\alpha) \frac{w(x, y)}{\sum_{z \in \Gamma(x)} w(x, z)} + \begin{cases} \alpha & \text{if } y \text{ is the central node} \\ 0 & \text{otherwise} \end{cases}.$$

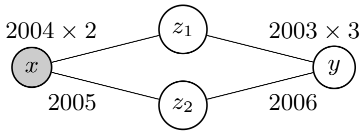


Figure 1: Edge-induced properties of paths. All papers have two authors, the labels of edges represent the time and the number of publications. x is the central node and it is connected with y by two paths. In the example we consider only time of collaborations and their numbers. The edge induced properties of the upper path are $\langle 2003, 2 \rangle$. The properties of the lower path are $\langle 2005, 1 \rangle$. Therefore, node y is assigned scores $\langle 2005, 2 \rangle$ with respect to the central node x.

In our experiments the local neighborhood is a connected subgraph, therefore we do not need to consider nodes that cannot be reached from v.

Edge-induced Properties of Paths. The properties of edges, which are used for repeated link prediction, induce properties of paths. We consider an example of the time of the last collaboration between authors. Let $p = \langle v_1, v_2, \ldots, v_{n+1} \rangle$ be a path of length n that connects v_1 and v_{n+1}. We can assign the year of the last collaboration to each pair of nodes along the path. The assignment defines a sequence of years $\langle y_{1,2}, y_{2,3}, \ldots, y_{n,n+1} \rangle$. The property of the path induced by edges is:

$$y(p) := \min_i y_{i,i+1}. \qquad (10)$$

If v is the central node in our node-centric testing method, than the score of node w is

$$score_v^y(w) := \max_{p \in \mathcal{P}(v,w)} y(p), \qquad (11)$$

where $\mathcal{P}(v, w)$ is the set of all shortest paths connecting v and w. We consider only the shortest paths since including all possible paths between two nodes is both inefficient and may result in blurring of information from shorter paths. Further, this choice also ensures that if two nodes are connected directly by an edge, the score from the generalized method will be the same as using the standard edge-property.

Exactly the same definition of generalized properties can be used to generalize the number of common collaborations between two authors to paths. The minimal number of co-authors, the property indicates stronger connection if it is smaller. Therefore, in the generalized version of the connection strength $\min$ in (10) becomes $\max$, and $\max$ in (11) becomes $\min$. Fig. 1 presents an example, where the generalized properties are calculated for paths of length 2.

4.2 Time-aware Maximum Entropy

Publications record author's interests and social connections. Both can change over time. Scientists find new research areas exciting and move between universities. Clearly, older publications are less relevant to link prediction than more recent ones. Our time-aware method works by allowing to forget old events, thus minimizing noise which they may introduce.

The events are forgotten when the constraints that they impose on P can be violated. In our version of maximum entropy the violations are penalized with weights w_j and a penalty term is added to the objective function. Changing the weights enables us to smoothly interpolate between the solution of standard maximum entropy ($\forall_j w_j = \infty$) and the prior distribution ($\forall_j w_j = 0$). Additionally, we can trade violation of one constraint for violation of another. The penalized maximum entropy is

$$\begin{aligned} \min_P \quad & D(P \parallel Q) + \sum_{j=1}^{m} w_j \beta_j && (12) \\ \text{s.t.} \quad & \left| \sum_{x \in S} P(x) k(x|j) - d(j) \right| \leq \beta_j \ \text{ for } j = 1, \ldots, m \\ & \beta_j \geq 0 \\ & \sum_{x \in S} P(x) = 1 \end{aligned}$$

4.2.1 Lagrange Multipliers for Time Aware Problem

We again use Lagrange multipliers to transform the time-aware problem. We obtain

$$P(x) = Q(x) \exp\left(-1 - \sum_j k(x|j)\lambda_j - \nu \right) \qquad (13)$$

The dual function is

$$g(\lambda, \nu) = \sum_x Q(x) \exp\left(-1 - \sum_j k(x|j)\lambda_j - \nu \right) + \sum d_j \lambda_j + \nu \qquad (14)$$

$$\lambda_j \in [-w_j; w_j] \qquad (15)$$

There is a close connection between time-agnostic and time-aware problem. The dual functions differ only in domains. In the time aware case it is restricted. Limited domain of λ restricts coefficients which multiply $Q(x)$. Therefore, the time-aware solution should be closer to the prior distribution Q.

Similar forms of dual problems allow us to use the same algorithm to solve both optimization problems. Our solvers use BLMVM algorithm [3] provided by Toolkit for Advanced Optimization [2].

4.3 Weights in Time-aware Method

Constraints in our optimization problem originate from publications. Therefore, a natural way to obtain weights for constraints is to derive them from some weights assigned to publications. We perform this in the following way:

1. Weight of a paper is a strictly increasing function of the time of its publication. The oldest and the latest publications are assigned weights w_{min} and w_{max} respectively. We require that $w_{min} \geq 0$ and $w_{max} \geq w_{min}$. In our experiments we used three functions. If t denotes the time of publication normalized in such way that the beginning of the data set corresponds to 0.0 and the end to 1.0, then the weighting functions are scaled and shifted variants of $\exp(3t)$, t and $\sqrt(t)$. The details are given in Section 6.4.1.

2. Weight of a constraint is the average or the sum of weights of the papers which contribute to the constraint. The average discards information about the number of papers and preserves only temporal information. On the other hand, the sum depends both on freshness of papers and the number of them. Let us call the weight of the jth constraint w_j.

3. Constraints are scaled according to the following formula:

$$w_j \leftarrow l_j \left(\frac{w_j}{\max_k w_k} \right)$$

where vector l is the solution of the dual, time-agnostic problem.

If we allowed wide enough domain in the time-aware problem ($\lambda_j \in [-|l_j|, |l_j|]$), the solution would be the same as the solution of the time-agnostic problem. Therefore, for some constraints the domains of the dual variables in the time-aware problem should be proper subsets of $[-|l_j|, |l_j|]$. This is achieved in the last step of assigning weights to constraints.

4.4 Numerical Solution

Maximum entropy models are usually fitted by specialized algorithms, e.g. *iterative scaling* [8]. Malouf compared [12] them with general purpose optimization algorithms. Surprisingly, the general purpose algorithms performed better. The best results were achieved by *limited memory variable metric* method (LMVM). In our experiments we fit maximum entropy with the BLMVM algorithm [3], which is a variant of LMVM, incorporating bounds on domains necessary for the time-aware maximum entropy.

5. EVALUATION FRAMEWORK

In our evaluation, we use co-authorship graphs which has time information associated with each (hyper)edge. We divide the datasets into two parts: (i) part that contains all data that was collected until year y, and the second part, i.e., the data collected after year y. The first part is used to derive features for link prediction (the training set), and the second part serves as the ground truth for testing the performance of prediction techniques.

5.1 Effectiveness Metrics

As we reasoned earlier, a link prediction method should be evaluated by choosing a vertex v and ranking its neighbors by the propensity that they will be linked to v in the future. As rankings are sequences of items, it is difficult to compare them objectively. Therefore, we need a method to assign them numerical scores, which reflect their quality. Comparisons of rankings may focus on the relevance of top-k elements or the overall quality. As such goals cannot be reconciled, we propose two quality metrics.

We use ground truth to assign binary labels to vertices. A vertex is labelled 1 if it is linked with the central node in the testing period, and 0 otherwise. A ranking of k items is a function $r : \{1, \ldots, k\} \to \{0, 1\}$. We denote it by$\langle r(1), \ldots, r(k) \rangle$.

Discounted Cumulative Gain (DCG)

The first method of evaluating a ranking of nodes is based on the standard *discounted cumulative gain* (DCG) used in information retrieval [13]. It is defined as:

$$\text{DCG}(r) = \sum_{i=1}^{k} \frac{r(i)}{\log_2(i+1)}.$$

The denominator grows with i, discounting contribution from items which are far from the front of the list. Therefore, DCG penalizes any wrong ordering at the beginning of the result list heavily.

While it is possible to use raw DCG scores to assess the quality of rankings, these are not suitable in our evaluation method due to the following reasons:

- DCG depends on the number of relevant items. This makes it impossible to compute a meaningful average of the DCG scores from rankings of different lengths or with different numbers of 1's. DCG values of such rankings are not comparable either.

- The expected DCG of a random ranking is some non-zero value, which also depends on the number of 1's and the length of the ranking. It means that in order to check if a method is better than a random ordering, we need to compare DCG with the expected DCG of a random ranking.

We solved the above issues by normalizing DCG. First, we calculate the expected value of $\mathrm{DCG}(r)$:

$$\mathbf{E}[\mathrm{DCG}(r)] = \mathbf{E}\left[\sum_{i=1}^{k} \frac{r(i)}{\log_2(i+1)}\right] = \sum_{i=1}^{k} \frac{\mathbf{E}\left[r(i)\right]}{\log_2(i+1)}$$
$$= \frac{|\{i : r\,(i) = 1\}|}{n} \cdot \sum_{i=1}^{k} \frac{1}{\log_2\,(i+1)}$$

We additionally need DCG values of the worst and the best ranking. In the best possible ranking all 1's occur before the 0's, the DCG of such ranking will be denoted by $\mathrm{DCG}_{best}(r)$, conversely the worst DCG will be denoted by $\mathrm{DCG}_{worst}(r)$ and it is the score of a ranking where all 0's occur before the 1's. We normalized DCG and the expected DCG, so they fall into $[0, 1]$ interval:

$$\mathrm{DCG}_{01}(r) := \frac{\mathrm{DCG}(r) - \mathrm{DCG}_{worst}(r)}{\mathrm{DCG}_{best}(r) - \mathrm{DCG}_{worst}(r)}$$
$$\mathrm{EDCG}_{01}(r) := \frac{\mathbf{E}[\mathrm{DCG}(r)] - \mathrm{DCG}_{worst}(r)}{\mathrm{DCG}_{best}(r) - \mathrm{DCG}_{worst}(r)}.$$

Our final DCG-based score measures how much the result is better than the expected result of a random ranking:

$$score_{\mathrm{DCG}}(r) := \frac{\mathrm{DCG}_{01}(r)}{\mathrm{EDCG}_{01}(r)} = \frac{\mathrm{DCG}(r) - \mathrm{DCG}_{worst}(r)}{\mathbf{E}\left[\mathrm{DCG}\,(r)\right] - \mathrm{DCG}_{worst}(r)}$$

The construction of the score makes the expected score of a random ranking equal 1. The upper bound depends on the length and number of 1's in the ranking, therefore it must be reported alongside with the score.

Average Normalized Rank (ANR)

We introduce the second measure which tells us directly where the relevant items are placed in the ranking. The measure weighs all positions equally, and therefore it is useful if we are also interested in the item placed behind the beginning of a ranking.

If the compared ranking contains only one item labeled 1 at a position i, the measure shows the fraction of the ranked items we need to skip to reach i, i.e. $(i - 1)/k$. If more such items exist we take the average of their scores:

$$score_{ANR}(r) := \frac{1}{k} \sum_{\{i:r(i)=1\}} \frac{i-1}{k}$$

Therefore the value of ANR is bounded to the $[0, 1]$ interval. The *expected* value of ANR measure is $(k - 1)/2k$, which is close to $\frac{1}{2}$. An ideal ranking will obtain a score close to 0, worst possible ranking will obtain a score close to 1.

Property	**DBLP**	**astro-ph**
time span	1997-2006	1997-2006
number of authors	437515	55233
number of publications	522932	60996
number of collaborations	1359471	644496
avg. authors per paper	3.08	4.72
avg. papers per author	3.69	5.21

Table 1: Summary of data sets.

6. EXPERIMENTS

6.1 Data Sets

We tested our link prediction methods on two data sets:

- the DBLP (`http://dblp.uni-trier.de/`) bibliographic database of computer science articles;

- the `astro-ph` (astrophysics) section of ArXiv (`http://arxiv.org/`), which is an archive of mostly physics preprints.

They are publicly available and were used in [11, 19]. Therefore it is possible to compare the performance of our methods with the prior works. Both the datasets contain publications spanning more than a decade. Unfortunately, the density of publications recorded over time is not consistent – with earlier years sparsely represented. We eliminated these early years, and used information of publications from the past 10 years. Note that, this data-cleaning is consistent with other work that have used the same sources of collaboration networks [19]. We further clean the dataset by eliminating publications written by only one author since they do not help in link prediction framework in anyway. The publication data from the first 8 years are used to build subgraphs on which the methods are tested. The remaining 2 years yield the ground truth, against which we compute the prediction accuracy using DCG and ANR measures we introduced earlier.

Table 1 summarizes various features of the data sets we have used. The number of collaborations is the number of pairs of authors who collaborated at least once, and it is equal to the number of edges in a simple graph of collaborations. We can see from the table, that the papers in DBLP data set have less authors, the average is 3.08 authors per paper compared with 4.72 for the `astro-ph`. Authors in DBLP have less papers than those from the `astro-ph` data set. Based on these observations we expect that distance-k neighborhoods in DBLP are less than in `astro-ph`. Table 3 confirms this conjecture.

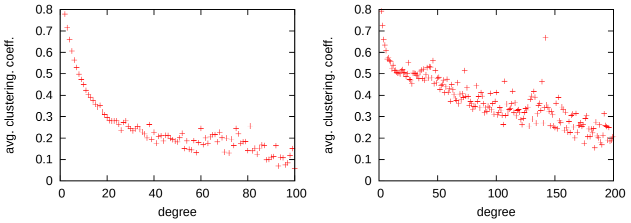
6.2 Feasibility of Link Prediction

Effectiveness of link prediction methods depends on the clustering coefficient of the network and the collaboration distance among new links. We present the results of our detailed analysis of DBLP and `astro-ph` networks on these two factors.

6.2.1 *Clustering coefficient*

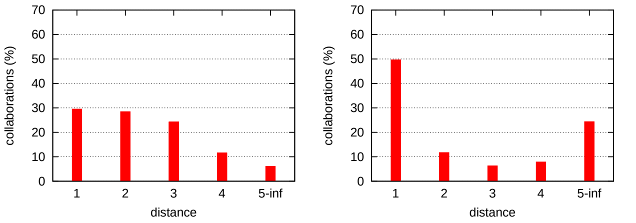
The clustering coefficient of a network is correlated with the probability that a triple of vertices will form a triangle. Watts and Strogatz define [20] the clustering coefficient of a vertex as:

$$C(v) = \frac{\text{number of triangles connected to } v}{\text{number of triples centered on } v}. \qquad (16)$$



(a) DBLP (b) astro-ph

Figure 2: Clustering coefficient as a function of degree.



(a) astro-ph (b) DBLP

Figure 3: Distance between collaborating authors. Last column contains collaboration at distance 5 and higher, including infinity (collaborations between authors who were not connected).

Instead of averaging $C(v)$ over all the vertices to get the clustering coefficient of a network, we can analyze it as a function of degree. We define the average coefficient for degree i as

$$C(i) = \frac{1}{|\{v \mid degree(v) = i\}|} \sum_{\{v|degree(v)=i\}} C(v). \qquad (17)$$

We omit two types of uninteresting vertices: those with less than two neighbors, since the clustering coefficient (16) is undefined for them, and vertices corresponding to authors with only one paper, because their neighborhoods form cliques and the clustering coefficient is always 1. The distribution of $C(i)$ is plotted in Fig. 2. The figure shows that the coefficient decreases with the degree of a node. Hence, if two nodes are connected by a node with a low degree, they are likely to be connected too. This does not hold if the intermediate node has high degree. The observation justifies the effectiveness of Adamic-Adar measure in link prediction.

6.2.2 *Distance Between Collaborating Pairs*

We observed that a large fraction of collaborations happen between authors who are *close* in the collaboration graph. For example, in DBLP, 50% of collaborations occur between authors who have already co-authored a paper in the past. Additional 12% of collaborations occur between authors who have at least one co-author in common. The distribution of distances between collaborating nodes is presented in Table 3. The fact that most links are created between nodes that are close in the graph was also observed by Leskovec et al. [9]. Furthermore, they show that the number of nodes at distance h from given node grows exponentially. Hence, as the distance between two (disconnected) nodes increases, it gets harder to predict their linkage in the future.

6.3 Testing of Link Prediction Methods

Our testing method ranks the neighborhood of a node v. We aim to choose the neighborhood which captures most of the collaboration of v. The results presented above show that such neighborhood can consist of direct neighbors of v and possibly also nodes which are at the distance 2 from v. Taking into account more neighbors does not improve the recall of collaborations much, but it significantly increases the size of the neighborhood.

In Section 2.2 we defined two link prediction tasks: prediction of both repeated and new links, and prediction of only new links. Repeated links are formed solely between nodes at distance 1. Hence:

- for prediction of *repeated and new links*, the neighborhoods of the central nodes consist of nodes at distances 1 and higher;
- for prediction of *new links only*, these neighborhoods consist of nodes at distances 2 and higher;
- for the task of *repeated link prediction*, the neighborhoods contain *only* the nodes at distance 1 from the central nodes.

6.3.1 *Local Neighborhood Selection*

We used maximum entropy methods to predict repeated link occurrences, i.e. neighborhoods contained only direct neighbors of central nodes. Additionally, we require that central nodes fulfill $4 \leq |\Gamma(v)| \leq 8$. The lower bound removes uninteresting cases. The upper bound is necessary for efficient calculations of maximum entropy methods. The maximum entropy methods model the probability of papers to occur. If a neighborhood has size n, then there are 2^n possible papers. Thus, maximum entropy has exponential complexity and this limits the size of the neighborhood that can be processed. We empirically found that the maximum entropy methods work fast enough when there are at most 8 nodes in the neighborhood plus the central node. Wang et al. [19] built maximum entropy models on neighborhoods of size 8, which is 1 less than in our experiments (8 neighbors + 1 central node). The problem with the complexity of maximum entropy methods was not evident in the edge-centric settings, where the subgraph connecting two nodes can be quite small. However, in the node-centric approach it means that we have to prune the neighborhood of the central node. We follow the simplest possible pruning algorithm: if a central node has more than 8 neighbors, we chose at random 8 of them. Moreover, we prune away neighbors that do not meet a minimum support threshold of having at least 5 papers.

For other link prediction methods, we test using the following types of neighborhoods: (i) direct neighborhoods, (ii) neighborhoods expanding to distance 2, (iii) neighborhoods expanding to distance 3. As they do not face scalability issues like the maximum entropy methods, we do not limit the size of neighborhoods. No thresholds on the number of papers are applied.

6.4 Experimental Results

6.4.1 *Performance of Maximum Entropy Methods*

Table 2 contains results of our experiments with time-aware and time-agnostic maximum entropy methods. The reported values are averages of DCG and ANR scores over sets of neighborhoods. We varied the parameters of the time-aware maxent. Rows marked with "avg." and "sum" describe which function is used to transform weights of papers into weights of constraints, we describe them in Section 4.3. Rows marked with "exp.", "lin." and "sqrt" show the

Method	DBLP		astro-ph	
	DCG	ANR	DCG	ANR
Best possible	2.5710	0.1020	2.5946	0.1178
ME	1.6728	0.2883	1.5819	0.3211
TME, avg., exp.	1.8473	0.2373	1.6368	0.2955
TME, avg., lin.	1.8685	0.2334	1.6698	0.2890
TME, avg., sqrt.	1.8713	0.2339	1.6825	0.2880
TME, sum, lin.	1.4804	0.3181	1.3289	0.3211
sort by count, last	1.7646	0.2546	1.6621	0.2945
sort by last, count	1.8823	0.2316	1.6937	0.2875
PageRank (PR)	1.2641	0.3824	1.2704	0.3882
PR (time)	1.5640	0.2998	1.4718	0.3337
PR (min. coauthors)	1.4876	0.3312	1.4513	0.3499
common neighbors	1.4377	0.3411	1.4271	0.3537
Jaccard	1.2430	0.3746	1.2597	0.3791
Adamic-Adar (AA)	1.5293	0.3198	1.4600	0.3460
AA (time)	1.6761	0.2843	1.5850	0.3151
AA (min. coauthors)	1.5830	0.3045	1.5814	0.3159
AA (count)	1.5161	0.3217	1.5025	0.3341
min. coauthors	1.4817	0.3059	1.3615	0.3475

Table 2: Results of experiments with repeated link prediction. ME is the (time-agnostic) maximum entropy, TME is the time-aware maximum entropy.

functions used as weights of the papers. The time of publication was normalized. That is, year 1997 corresponds to 0.0, and 2004 to 1.0. Let y be the time a publication appeared with respect to the normalization. Possible weights of the paper are: $\exp(3y)$, y and $sqrt(y)$ scaled and shifted such that the weight of a paper from 1997 is $w_{min} = 0.2$ and the weight of a paper from 2004 is $w_{max} = 1.0$.

The results show that time-aware methods outperform the time-agnostic ones. The improvement is significant and happens consistently for all methods extended with temporal information, on both data sets. The results are confirmed by both DCG and ANR metrics. The improvement of DCG, means that temporal information positively influences the quality of the top of rankings, which is important in applications where the results are presented to a human user. The simplest method – "last, count", which sorts the nodes by the time of the most recent collaboration with the central node and the number of collaborations, shows slightly better performance than the time-aware maxent. It also outperforms the other presented methods. We observe that varying the choice of a function which assigns weights to papers does not influence the performance of maxent much. However, the way the weights of papers are transformed into weights of constraint does. "sum" which captures both temporal information and frequency of events, yields weaker results in comparison with "avg." which captures only time of publication of papers.

6.4.2 *Performance of Baseline Methods and Their Extensions*

The results of experiments with baseline link prediction methods are presented in Tables 3 and 4. Table 3 presents the performance of methods on the task of predicting new and repeated links. Table 4 shows the performance of predicting new links only. *Distance* means ordering nodes by the distance from the central node. Predictors based on *last* and *count* are undefined for nodes at distances higher than 1; such nodes are placed at the end of rankings. We also used this strategy for other predictors that are undefined for high distances. For Adamic-Adar and rooted PageRank, we tested standard and weighted versions. Time weighting was done with the "lin." function used in the maximum entropy methods.

"Last, count" and time-aware achieves the best results in prediction at distance 1. Because DCG evaluation weights entries at the top of ranking the heaviest, and most of new collaborations are at distance 1, the results carry over into prediction at higher distances. This happens even though both methods are not able to handle distances higher than 1. On the other hand, ANR weights all positions equally, thus it is sensitive to quality of ranking for higher distances. The effect can be observed when we compare PageRank and "last, count" for astro-ph with distances up to 3. Even though DCG scores are similar, ANR scores show that PageRank is preferred.

The edge-induced properties of paths are marked with "gen.", since they generalize methods such as "last". They are used in combination with distance in the graph. For example the "dist., last, count (gen.)" method sorts the nodes by their distance from the central node, the generalized time of the last collaboration and the generalized number of common collaborations. Table 3 shows that the edge-induced properties of paths achieve good performance, however this effect comes from the good performance at distance 1 – experiments with predicting new links show that the edge-induced properties are inferior to other methods.

We observed that "last, count" is effective in predicting repeated links, whereas time-aware rooted PageRank has high performance at higher distances. The two methods can be easily combined: at the top of the ranking we put nodes at distance 1 ordered by "last, count", and we order the remaining nodes by the other method. The combination is marked as "last, count + PR (time)".

The results show that link prediction methods can be improved by taking time into account. For repeated link prediction methods that consider only temporal information can easily outperform methods based on the network structure. Results for new link prediction (Table 4) show that we are far from the best possible score. This should be expected, because link prediction is a difficult task. The difficulty is also visible in results from other authors, for example Liben-Nowell and Kleinberg [11]. Even though the best method outperforms the random classifier by the factor of 18.0 (on the `astro-ph` data set), its precision is only $18.0 \times 0.475\% = 8.56\%$.

7. RELATED WORK

The information needed for link prediction comes from two sources: the topology of the networks and attributes of nodes. In case of co-authorship attributes of nodes can be affiliations of the authors, abstracts of their publications, etc. The topological data can be used alone. Liben-Nowell and Kleinberg conducted [11] a comprehensive study of different proximity measures. They discovered that methods based on proximity measures can outperform a random classifier by factor of forty to fifty. Still, the precision of their methods is low. Wang et al. [19] combine distance measures, nodes' attributes and a novel feature based on a local probabilistic model of the network. We used their model as the base for our time aware method. O'Madadhain et al. [16] view link prediction as a binary classification problem. They use the history of events in the network and attributes of nodes as input to a logistic regression classifier, which decides whether two nodes will be linked in future. Apart from link prediction, they also discuss temporal ranking of

Data set	DBLP						astro-ph					
Distance	1		1 and 2		1, 2 and 3		1		1 and 2		1, 2 and 3	
Avg. $\vert\mathcal{N}(v) \cap \{v\}\vert$	14		106		867		36		758		5498	
	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR
best possible	2.9601	0.1016	7.6964	0.0257	17.7665	0.0085	3.2480	0.1266	11.4819	0.0197	23.1410	0.0051
random classifier	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5
distance	1.0097	0.4355	3.9026	0.1566	8.5377	0.0869	0.9910	0.4667	4.1090	0.2411	9.6214	0.1879
PageRank (PR)	1.4817	0.3835	4.7840	0.1233	9.9678	0.0740	1.6088	0.3956	5.2472	0.1611	11.4039	0.1120
PR (time)	1.7431	0.3101	5.1837	0.1088	10.9626	0.0627	1.7797	0.3447	5.6979	0.1523	12.1684	0.1084
PR (min. coauthors)	1.6972	0.3335	5.0994	0.1129	10.9155	0.0670	1.8348	0.3432	5.8372	0.1476	11.8490	0.1026
dist., last, count (gen.)	1.9537	0.2621	5.3173	0.1087	11.6286	0.0629	1.8853	0.3228	5.7883	0.1836	12.4530	0.1459
dist., count, last (gen.)	1.8395	0.2807	5.1757	0.1099	11.4080	0.0665	1.8484	0.3280	5.7113	0.1841	11.9414	0.1447
dist., min. coauth. (gen.)	1.5122	0.3309	4.7849	0.1205	10.1923	0.0714	1.5043	0.3730	5.2794	0.1847	11.3909	0.1353
last, count + PR (time)	1.9537	0.2621	5.3213	0.1077	11.6763	0.0597	1.8853	0.3228	5.9317	0.1516	12.8994	0.1044
common neighbors‡	1.4436	0.3880	3.1171	0.3232	7.0728	0.1651	1.5214	0.4081	4.4464	0.2552	10.0292	0.2003
Jaccard‡	1.0677	0.4565	2.4980	0.3642	6.3979	0.1771	1.1444	0.4626	3.7287	0.2754	8.8862	0.1996
AA‡	1.5800	0.3618	3.7450	0.2702	8.2070	0.1568	1.6164	0.3894	4.7534	0.2205	10.2071	0.1981
AA (time)‡	1.7339	0.3248	4.0138	0.2383	8.9810	0.1524	1.7338	0.3560	5.0273	0.2036	10.8486	0.1953
AA (min. coauthors)‡	1.6314	0.3458	3.8474	0.2444	8.1427	0.1588	1.7616	0.3571	5.1396	0.1986	10.7729	0.1971
AA (count)‡	1.5799	0.3566	3.7606	0.2486	7.9819	0.1598	1.6318	0.3772	4.7985	0.2116	10.6964	0.1976
last, count†	1.9537	0.2621	5.2424	0.1384	10.9772	0.1412	1.8853	0.3228	5.6215	0.2338	11.3563	0.2833
count, last†	1.8395	0.2807	5.1032	0.1394	10.7972	0.1422	1.8484	0.3280	5.5513	0.2341	10.8170	0.2833
min. coauthors†	1.5122	0.3309	4.7306	0.1445	9.7276	0.1448	1.5041	0.3732	5.0730	0.2367	10.3519	0.2833

Table 3: Results of experiments with baseline methods. † (‡) methods undefined or zero for distance higher than 1 (2).

Data set	DBLP						astro-ph					
Distance	2		3		2 and 3		2		3		2 and 3	
	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR	DCG	ANR
best possible	11.3931	0.0079	30.8981	0.0006	25.6495	0.0019	15.7977	0.0083	36.4565	0.0016	31.5615	0.0021
random classifier	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5	1.0000	≈ 0.5
distance	1.0826	0.4882	1.0196	0.4926	2.7233	0.2333	1.0710	0.4963	0.9739	0.5128	2.8517	0.3000
PageRank (PR)	2.3930	0.3283	3.0785	0.2848	4.9076	0.1626	2.7659	0.2909	3.7092	0.2633	8.0563	0.1654
PR (time)	2.7467	0.2817	3.3069	0.2605	5.5343	0.1533	3.0023	0.2698	3.6966	0.2517	9.1074	0.1600
PR (min. coauthors)	2.6285	0.2939	3.3823	0.2312	5.2713	0.1413	2.9998	0.2610	3.8485	0.2371	9.2260	0.1520
dist., last, count,	2.3105	0.2919	2.6858	0.3292	5.6115	0.1764	2.1128	0.3535	1.8629	0.3635	6.7457	0.2220
dist., count, last	2.3334	0.2930	1.6180	0.3564	4.9558	0.1858	2.1941	0.3533	1.8184	0.3631	5.5872	0.2201
dist., min. coauth.	1.7605	0.3405	2.1357	0.3602	3.2593	0.1907	2.1344	0.3505	1.8582	0.3407	4.2890	0.2050
common neighbors‡	2.1369	0.3891	-	-	4.1443	0.2242	2.6044	0.3471	-	-	7.3016	0.2921
Jaccard‡	1.8300	0.3960	-	-	3.9664	0.2309	2.1812	0.3766	-	-	5.1899	0.2938
AA‡	2.2969	0.3486	-	-	4.7202	0.2223	2.7402	0.3125	-	-	7.5472	0.2908
AA (time)‡	2.8047	0.2667	-	-	5.5419	0.2196	3.0820	0.2862	-	-	8.0662	0.2894
AA (min. coauthors)‡	2.6694	0.2760	-	-	4.5814	0.2199	3.1340	0.2824	-	-	8.0622	0.2893
AA (count)‡	2.7111	0.2911	-	-	5.0897	0.2226	2.8808	0.2963	-	-	8.1917	0.2895

Table 4: Experiments with prediction of new links. ‡ methods undefined or zero for distance higher than 2.

nodes' importance. Murata and Moriyasu [14] study potential advantages of exploiting multiple edges available in some data sets. A multigraph representing a network is transformed into a simple graph, where the *weight* of an edge is the number of corresponding edges in the multigraph. Results show improvement over non-weighted methods. Clauset et al. [5] use hierarchical random graph models for link prediction. Nodes of a graph are leaves of a tree, which represents a recursive structure of communities. The method was used to infer missing links in: a metabolic network,a terrorist association network and a species interaction network. The results show that the new methods performed better than Jaccard coefficient and common neighbors. Recently, [7] used time series to perform time-aware link prediction. They are mostly interested in communication network surveillance. It such scenario events, like emails or phone calls, may trigger another events. In their experiments they demonstrate that time aware methods compare favorably with commonly used time unaware methods. The best results can be achieved by a combination of both. Sharan and Neville [18] followed a two-step approach to time-aware link prediction. First they summarize the dynamic graph with a weighted static graph, then they use the relational Bayes classifier. The experiments are performed on a heterogeneous data set of scientific collaborations and citations. Potgieter et al. [17] studied temporal metrics in link prediction. Among other methods, they used moving averages and percentage of change of baseline metrics, such as common neighbors or Katz. Moreover, they used Dynamic Bayesian Networks to mine the relationships between metrics and formation of links.

8. CONCLUSIONS AND FUTURE RESEARCH

In this work, we presented methods to incorporate temporal information available on evolving social networks for link prediction tasks. We showed that time of interactions between entities is a dominant feature for ranking neighboring nodes based on their probability of future interaction with the central node. This observation holds beyond the immediate direct neighborhood of the chosen node. A very simple time-aware method of (last,count)-based ranking is shown to outperform more other, more sophisticated, techniques in repeated link prediction. The performance of simple baseline predictors is improved by time-based weighting of edges. We also shown, that the performance of predictors is improved by weighting edges according to the connection strength between the authors.

An important contribution of our work is a novel node-centric approach to the evaluation of link prediction. Our node-centric testing consist in ranking a set nodes close to a central node v according to the probability that they will collaborate with v. It resembles usage of real world link prediction systems, such as the *People You May Know* tool on Facebook.

In the future we would like to test our link prediction methods on data sets other than scientific collaboration networks. We also would like to further investigate combining features for ranking nodes. Finally, we would like to look closer at the problem of disambiguating entities in the network. Both entity disambiguation and link prediction can use structure of the network. Therefore, there is a possibility of applying observations or methods from one field to the other.

9. REFERENCES

[1] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211 – 230, 2003.

[2] S. Benson, L. C. McInnes, J. Moré, T. Munson, and J. Sarich. *TAO User Manual (Revision 1.9)*, 2007. http://www.mcs.anl.gov/tao.

[3] S. J. Benson and J. Moré. A limited-memory variable-metric algorithm for bound-constrained minimization. Technical Report ANL/MCS-P909-0901, Mathematics and Computer Science Division, Argonne National Laboratory, 2001.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

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

[6] The DBLP computer science bibliography. `http://www.informatik.uni-trier.de/~ley/db`.

[7] Z. Huang and D. Lin. The Time-Series Link Prediction Problem with Applications in Communication Surveillance. *INFORMS Journal on Computing*, 2008.

[8] F. Jelinek. *Statistical methods for speech recognition*. MIT Press, Cambridge, MA, USA, 1997.

[9] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470, New York, NY, USA, 2008. ACM.

[10] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 915–924, New York, NY, USA, 2008. ACM.

[11] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[12] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[14] T. Murata and S. Moriyasu. Link prediction of social networks based on weighted proximity measures. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 85–88, Washington, DC, USA, 2007. IEEE Computer Society.

[15] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003.

[16] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *ACM SIGKDD Explorations Newsletter*, 7(2):23–30, 2005.

[17] A. Potgieter, K. April, R. Cooke, and I. Osunmakinde. Temporality in Link Prediction: Understanding Social Complexity. Sprouts: Working Papers on Information Systems, 7(9), 2007.

[18] U. Sharan and J. Neville. Exploiting time-varying relationships in statistical relational models. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 9–15. ACM New York, NY, USA, 2007.

[19] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 322–331, 2007.

[20] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.