A Parameterized Probabilistic Model of Network Evolution for Supervised Link Prediction

Hisashi Kashima
IBM Research
Tokyo Research Laboratory
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502, Japan
hkashima@jp.ibm.com

Naoki Abe
IBM Research
T.J.Watson Research Center
1101 Kitchawan Road and Rte. 134
Yorktown Heights, NY 10598
nabe@us.ibm.com

Abstract

We introduce a new approach to the problem of link prediction for network structured domains, such as the Web, social networks, and biological networks. Our approach is based on the topological features of network structures, not on the node features. We present a novel parameterized probabilistic model of network evolution and derive an efficient incremental learning algorithm for such models, which is then used to predict links among the nodes. We show some promising experimental results using biological network data sets.

1 Introduction

For modeling and analyzing various phenomena concerning a collection of entities, it is rarely sufficient to examine the properties of the entities themselves. The essence of their collective behavior is often embedded within the relationship among them, such as co-occurrence, causality, and other types of interactions. Such relations can, for the most part, be represented as a *network* consisting of a set of entities and links between them.

Examples of network data around us that call for analysis are abundant. The world wide web consists of pages and hyperlinks among them. Social Networks consist of individuals and relations among them, such as friendship. They are attracting considerable attention from various business perspectives, such as marketing and business process modeling. In the field of bioinformatics, network structures among biological entities such as genes and proteins, representing physical interactions and gene regulation, are studied extensively. (See Figure 1 for an example biological network.) Links among entities are not limited to static relations, but may vary dynamically. For example, e-mail exchanges and cooperative interactions are temporal relations. In the field of sociometry, there has been a long history of social network analysis, in which relations among social entities are analyzed [26].

With the large amount of network data becoming readily available in electronic form today, along with the advances in computing power and algorithmic techniques that enable handling of such massive data, there has been a surge of interest in the study of analytical methods for network structured data, and *link mining* [6] has become a popular subarea of data mining.

Link mining includes several tasks such as link-based object classification/ranking/clustering, link prediction, and subgraph discovery [6]. In the present paper, we consider the link prediction problem, which is the task of predicting unobserved portion of the network i.e. hidden links, from the observed part of the network (or to predict the future structure of the network given the current structure of the network.)

Link prediction has several applications including predicting relations among participants such as friendship and pecking order, and predicting their future behavior such as communications and collaborations. In the field of bioinformatics, predicting protein-protein interactions and regulatory relationships can provide guidance on the design of experiments for discovering new biological facts.

The link prediction problem is usually formalized as a binary classification problem or a ranking problem on the node pairs. There are two types of information that may be useful for this objective – information on the nodes themselves and that on the network topology. There have been several studies of link prediction with node features, but comparatively little work has been done with topological features [21, 7, 23, 20]. In this paper, we focus on the latter.

Topological features can be derived from generative models of network structures [18]. Perhaps the most well-known model is the preferential attachment model proposed by Barabási et al. [2]. Liben-Nowelly and Kleinberg proposed and compared a number of metrics derived from various network models [15]. In this paper, we build on these earlier works and propose a novel parameterized model for network evolution, which is then used to derive a method of link prediction. Our work differs from these earlier works in that the existence of tunable parameters within the network model naturally gives rise to a *learning* algorithm for link prediction, leading to improved accuracy of prediction.

In our model, probabilistic flips of the existence of edges are modeled by a certain "copy-and-paste" mechanism between the edges. Our link prediction algorithm is derived by assuming that the network structure is in a *stationary* state of the network. This allows us to formalize the inference of the stationary state as a transduction problem, and propose an Expectation-Maximization (EM)-based transduction method [19]. The algorithm embodies a maximum likelihood estimation procedure using exponentiated gradient ascent [8].

We evaluate the effectiveness of the proposed approach by empirically comparing its predictive performance with link prediction methods based on the various topological metrics mentioned above. The results of our experiments indicate that the predictive performance of the proposed method, in terms of precision recall curves, significantly out-performs these existing methods, for the biological network datasets used in our experiments.

The rest of the paper is organized as follows. In Section 2, we introduce our definition of link prediction problem. We propose a probabilistic model of network structure evolution in Section 3, and a link prediction method based on this model, in Section 4. In Section 5, we review related work. In Section 6, we show results of our experiments. We conclude this paper with some discussion of future work in Section 7.

2 Link prediction problem

We start with defining the link prediction problem we consider in this paper.

Let the data domain be represented as a graph $G = (V, \phi)$, where $V = \{1, 2, \dots, |V|\}$ is the set of indices of nodes, and $\phi : V \times V \to [0, 1]$ is an *edge label* function. Each node $v \in V$ represents an entity, for example, a person in the case of social networks, and a protein in the case of protein-to-protein interaction networks. An *edge label* $\phi(i, j)$ indicates the probability that an edge exists between any pair of nodes. In particular, $\phi(i, j) = 1$ if an edge exists between i and j, and $\phi(i, j) = 0$ if an edge does not exist between i and j. Note that ϕ is symmetric.

Let $E^L \subset V \times V$ be the *labeled pairs*, which is a set of pairs of nodes whose edge labels are known. They will serve as the training data set for our problem. The *Link prediction problem* is the task of predicting the values of edge labels for $E^U := (V \times V) - E^L$, given V and E^L as training data. Example instances of the link prediction problem include that of predicting potential links such as friendship relations in social networks, or interactions among proteins in protein networks.

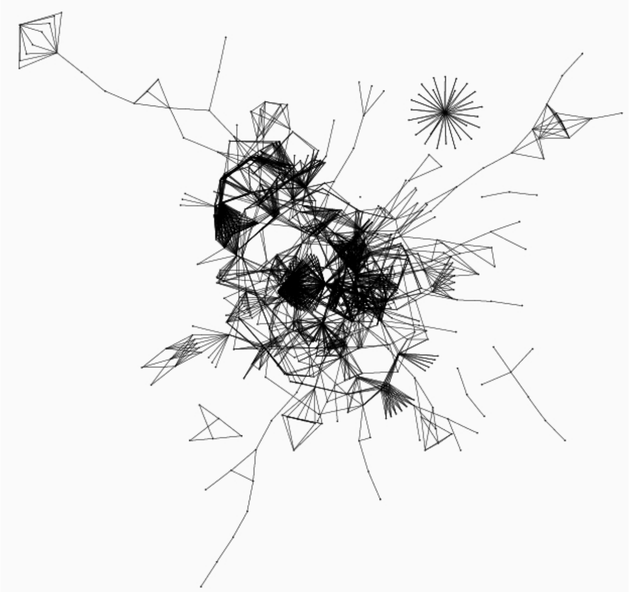

Figure 1. A metabolic network of S. Cerevisiae. Proteins are represented as nodes, and an edge represents catalyzation of successive reactions by two proteins.

Note that, in our setting, we do not assume the existence of any features on the nodes that can be used for inference. For example, there may be information about each person in a social network, such as the name, age, hobbies, etc. Some existing work in the literature has used such features for link prediction. Certainly, the ultimate goal would be to utilize all available information that may be helpful for prediction. In this paper, however, we focus on the use of topological features having to do with the structure of the network, and consider the link prediction problem based solely on them.

3 An edge label copying model of network structure evolution

In this section, we propose a parameterized probabilistic model of network evolution, in which the structure of a network probabilistically changes over time. Based on this model, we predict whether an edge exists between two nodes or not, at any given point in time. By the "structure of network" we mean the edge label function ϕ that indicates the existence of edges among nodes.

We denote by $\phi^{(t)}$ the edge label function at time t. We assume that only $\phi^{(t)}$ changes over time, and V, the members in the network, are fixed at all times, as we are primarily interested in the link structure among them.

In our model, we model probabilistic flips of the value of ϕ. Flips of the edge labels do not occur uniformly at random, but occur depending on some characteristics of the network. We characterize this by the process of "copy-and-paste" of edge labels between the nodes. The probability of copying differs from one node pair to another. We assume a Markov model in which $\phi^{(t+1)}$ depends only on $\phi^{(t)}$.

In the proposed model, an edge label is copied from node ℓ to node m randomly with probability $w_{\ell m}$. Once it has been decided that an edge label is to be copied from node ℓ to m, node ℓ copies one of its $|V|-1$ edge labels (excluding $\phi^{(t)}(\ell, m)$) to node m uniformly at random. We have the following probability constraints.

$$\sum_{\ell m} w_{\ell m} = 1,\ w_{\ell m} \geq 0. \tag{1}$$

Also, we assume that $w_{\ell\ell} = 0$, which indicates copying from a node to the node itself does not occur. Let W denote the matrix whose (ℓ, m)-th element is $w_{\ell,m}$.

The basic idea behind our model is as follows; if you have a friend who has a strong influence on you, your association will be highly affected by the friend's association. Also, if a gene is duplicated in the course of genetic evolution, the copied gene will have similar characteristics to the original one. Assume that node k has a strong influence on node i, and there is an edge between node k and node j. Following the above hypothesis, there will likely be an edge between node i and node j. Similarly, if there are no edges between k and j, there will likely be no edge between i and j.

There are two possible ways for $\phi^{(t)}(i,j)$ to assume a particular edge label. One possibility is that node k has copied an edge label to node i or to node j. The other is that $\phi^{(t)}(i,j) = \phi^{(t+1)}(i,j)$ and nothing has happened (indicating that a copy happened somewhere else in the network).

Following the above discussion, $\phi(i,j)^{(t+1)}$, the probability of an edge existing between node i and node j at time $t+1$, can be written as

$$\phi^{(t+1)}(i,j) = \frac{1}{|V|-1}\Big(\sum_{k\neq i,j} w_{kj}\phi^{(t)}(k,i) + w_{ki}\phi^{(t)}(k,j)\Big) + \Big(1 - \frac{1}{|V|-1}\sum_{k\neq i,j} w_{kj} + w_{ki}\Big)\phi^{(t)}(i,j), \tag{2}$$

where the first term indicates the probability that the edge label for (i,j) is changed by copy-and-pasting, and the second term indicates the probability that the edge label for (i,j) is left unchanged since copy-and-pasting happened somewhere else. In the first term, node k must have an edge to node j at time t (i.e. $\phi^{(t)}(k,j) = 1$) for node k to make node i span an edge to node j. Similarly, node k must have an edge to node i (i.e. $\phi^{(t)}(k,i) = 1$) for node k to make node j span an edge to node i. We assume that the edge label for (i,j) cannot be copied from node i to node j, and vice versa.

By iterative applications of this flipping equation of the edge labels, the network structure evolves over time.

4 Link prediction

In this section, we propose a new approach to link prediction based on the network structure evolution model we introduced in the previous section, and also propose a parameter estimation method from partially observed edge labels.

4.1 Stationary state of the network structure evolution model

Since we do not know the true parameters in (2), and do not in general observe the history of evolution of the network structure, it is unlikely that we can predict the existence of edges just based on this model. Therefore, we make an additional assumption that the current network that we are seeing is in some sense "typical" of the network structure averaged over time [10]. More precisely, we assume that it is a stationary state of the network evolution.

If the network structure is in a stationary state, by setting $\phi^{(\infty)}(k,i) := \phi^{(t+1)}(k,i) = \phi^{(t)}(k,i)$ in (2), we obtain

$$\phi^{(\infty)}(i,j) = \frac{\sum\limits_{k\neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)}{\sum\limits_{k\neq i,j} w_{kj} + w_{ki}}. \tag{3}$$

There are trivial solutions for this equation such as $\phi^{(\infty)}(i,j) = 0$ for all (i,j), or $\phi^{(\infty)}(i,j) = 1$ for all (i,j). As we will see in the next subsection, we constraint $\phi^{(\infty)}(i,j)$ to their actual values for (i,j) in training data E^L to obtain a nontrivial solution.

4.2 Parameter estimation and link prediction

We define the objective function for estimating the model parameters and determining the values of $\phi^{(\infty)}(i,j)$ for (i,j) in E^U. Since the log-likelihood, or cross entropy, for the edge label $\phi(i,j)$ is written as

$$L_{ij} = \phi^{(\infty)}(i,j) \log \left(\frac{\sum\limits_{k \neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}} \right)$$
$$+(1-\phi^{(\infty)}(i,j)) \log \left(1 - \frac{\sum\limits_{k \neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}} \right).$$

The total log-likelihood for the known edge labels E^L is defined as

$$L(W) = \sum_{(i,j) \in E^L} L_{ij}.$$

Our estimation problem is formulated as the following constrained optimization problem:

$$\text{Maximize}_{W, \phi^{(\infty)}(i,j) \text{ for } (i,j) \in E^L} \ L(W)$$
$$\text{such that}$$
$$\phi^{(\infty)}(i,j) = \frac{\sum\limits_{k \neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}}$$
$$\text{for } (i,j) \in E^U,$$
$$\text{and } \sum_{\ell,m} w_{\ell m} = 1, \ w_{\ell m} \geq 0.$$

Here, the objective function is the total log-likelihood for the known edge labels E^L, and the constraint is the condition for stationarity for the test data E^U. Since our problem is a transduction problem where the test data are given beforehand, not only the model parameters for copy-and-paste, but also the unknown edge labels $\{\phi^{(\infty)}(i,j) | (i,j) \in E^U\}$ are parameters to be determined. We employ an Expectation Maximization (EM)-type transductive learning approach [19] where optimization proceeds with respect to W and $\{\phi^{(\infty)}(i,j) | (i,j) \in E^U\}$, iteratively.

First, we describe the E-step of the transduction procedure. The hidden variables are $\phi^{(\infty)}(i,j)$ for $(i,j) \in E^U$ in our case. As we have seen in the previous subsection, they must satisfy Equation (3). Therefore, we solve the simultaneous linear equations to get the expected value of the hidden variables.

Next, we consider the M-step, which is to maximize L with all $\{\phi^{(\infty)}(i,j) | (i,j) \in E^U\}$ fixed.

This optimization problem is nonlinear, and does not have closed form solution. Therefore, we employ a gradient-based optimization to obtain a solution numerically. The gradient of the objective function with respect to $w_{\ell m}$ becomes

$$\frac{\partial L(W)}{\partial w_{\ell m}} = \sum_{i,j} \frac{\partial L_{ij}}{\partial w_{\ell m}},$$

where

$$\frac{\partial L_{ij}}{\partial w_{\ell m}} =$$
$$\phi^{(\infty)}(i,j) \cdot \left(\frac{\delta(m=j)\phi^{(\infty)}(\ell,i) + \delta(m=i)\phi^{(\infty)}(\ell,j)}{\sum\limits_{k \neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)} \right.$$
$$\left. - \frac{1}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}} \right)$$
$$+ (1-\phi^{(\infty)}(i,j))$$
$$\cdot \left(\frac{\delta(m=j)(1-\phi^{(\infty)}(\ell,i)) + \delta(m=i)(1-\phi^{(\infty)}(\ell,j))}{\sum\limits_{k \neq i,j} w_{kj}(1-\phi^{(\infty)}(k,i)) + w_{ki}(1-\phi^{(\infty)}(k,j))} \right.$$
$$\left. - \frac{1}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}} \right).$$

We employ the exponentiated gradient algorithm [8] since it ensures positivity of all elements of W and the probability constraint (1), and converges fast in the case of sparse W, which is the case for most of the real world networks.

Suppose that we have parameters W at a particular time of iterations, we want to update W to W' to increase $L(W')$. Also, the new W' is chosen to be sufficiently close to W by making the distance penalty $-d(W', W)$. We choose the distance as the relative entropy,

$$d(W', W) = \sum_{\ell,m} w'_{\ell m} \log \frac{w'_{\ell m}}{w_{\ell m}},$$

which leads to the exponentiated gradient algorithm.

At each iteration step, we determine the new W' that maximizes the following objective function with respect to (1).

$$\eta L(W') - d(W', W), \ \eta > 0,$$

where $\eta > 0$ is a constant that balances the two terms. Using Lagrangean multiplier γ, the objective function we wish to maximize is

$$F(W') \quad := \quad \eta L(W') - d(W', W) - \gamma \Big(\sum_{\ell,m} w_{\ell m} - 1 \Big)$$

Approximating $L(W')$ by using $L(W)$ as

$$L(W') = L(W) + \sum_{\ell,m} \frac{\partial L(W)}{\partial w_{\ell m}} (w'_{\ell m} - w_{\ell m}),$$

and setting the gradient of $F(W')$ with respect to $w'_{\ell m}$ to be zero, the Lagrangean that we maximize is

$$\frac{\partial F(W')}{\partial w'_{\ell m}} = \eta \frac{\partial L(W)}{\partial w_{\ell m}} - \Big(\log \frac{w'_{\ell m}}{w_{\ell m}} + 1 \Big) + \gamma = 0.$$

Solving this equation with $\sum_{\ell,m} w_{\ell m} = 1$ in mind, we obtain the following exponentiated gradient update,

$$w'_{\ell m} = Z^{-1} w_{\ell m} \exp\Big(\eta \frac{\partial L(W)}{\partial w_{\ell m}}\Big), \tag{4}$$

where

$$Z := \sum_{\ell,m} w_{\ell m} \exp\Big(\eta \frac{\partial L(W)}{\partial w_{\ell m}}\Big).$$

The above discussion is summarized as the two step procedure described in Figure 2. [Step:3] corresponds to the E-step, and [Step:4] corresponds to the M-step.

4.3 Scaling up the estimation algorithm by sequential updates

The transduction algorithm presented in the previous subsection runs in a batch manner, so can be somewhat inefficient for large data sizes. Therefore, we approximate this by a sequential version of the learning algorithm.

We perform the iteration of [Step:3] and [Step:4] in Figure 2 not with whole data, but with only one datum at a time. Assume that (i, j) is the node pair that we currently focus on.

As for [Step:3], instead of solving (3) as simultaneous linear equations, we process only one step of the power method for one unobserved edge label by

$$\phi^{(\infty)'}(i,j) := \frac{\sum\limits_{k \neq i,j} w_{kj}\phi^{(\infty)}(k,i) + w_{ki}\phi^{(\infty)}(k,j)}{\sum\limits_{k \neq i,j} w_{kj} + w_{ki}}. \tag{5}$$

Similarly, as for [Step:4], instead of (4), we employ stochastic approximation which processes one datum at a time,

$$w'_{\ell m} = Z^{-1} w_{\ell m} \exp\Big(\gamma \frac{\partial L_{ij}}{\partial w_{\ell m}}\Big) \tag{6}$$

for $\{(\ell, m) | \ell \in \{1, 2, \ldots, |V|\}, m \in \{i, j\}\}$, where γ is the coefficient for stochastic approximation. Note that Z includes the summation of $w_{\ell m}$ over all (ℓ, m), but we only compute the differences of Z, so each update is done in $O(|V|)$ time.

Also we execute the two steps in parallel. We randomly pick an (i, j) pair, and execute a micro E-step (5) if $(i, j) \in E^U$, and execute a micro M-step (6) if $(i, j) \in E^L$.

The description of the sequential version of the transduction algorithm is exhibited in Figure 3.

5 Experiments

In this section, we report on the experiments we conducted to compare the proposed method with some existing methods based on the topological features, using two biological network datasets.

5.1 Review of comparison methods

Before describing the experimental setting, we review the existing methods from earlier works [15, 28] we compare against the proposed method. These methods are based on node similarity metrics derived from certain network evolution models. Since each of these metrics described below quantifies the potential of an edge existing between a pair of nodes, they give rise to a *ranking* over node pairs and thus can be directly used to perform link prediction. (Link predictions can be given as ranking over the set of all possible edges.)

Note that all metrics defined below consider only the positive edge labels. In the definitions to follow, we let $\Gamma(i)$ denote the set of neighbor nodes connected to node i with edge label 1.

- Common neighbors [17]

  $$\text{common} := |\Gamma(i)| \cap |\Gamma(j)|$$

  Common neighbors metric is based on the idea that if two nodes i and j have many common neighbor nodes, they are likely to be linked.

- Jaccard's coefficient [3, 15]

  $$\text{Jaccard's} := \frac{|\Gamma(i)| \cap |\Gamma(j)|}{|\Gamma(i)| \cup |\Gamma(j)|}$$

  Jaccard's coefficient is a normalized version of the common neighbors metric, and is used as a similarity metric in the field of information retrieval. In Jaccard's coefficient, if two pairs of nodes have the same number of shared neighbor nodes, since in some sense their links are considered more "precious".

- Adamic/Adar [1]

  $$\text{Adamic/Adar} := \sum_{k \in |\Gamma(i)| \cap |\Gamma(j)|} \frac{1}{\log |\Gamma(k)|}$$

  Adamic/Adar metric is a variant of the common neighbors metric. The idea is similar to Jaccard's coefficient in the sense that a link owned by nodes with a smaller number of neighbors is considered more important, but Adamic/Adar assigns different weights among the neighbors. Common neighbors with a small number of neighbors are weighted more highly.

- Katz$_\beta$ [12]

  $$\text{Katz}_\beta := \sum_{l=1}^{\infty} \beta^l |\text{paths}_{i,j}^{(l)}|$$

  Katz$_\beta$ is a generalization of the common neighbors metric to account for more distant relations. Here

```
Algorithm: Batch
[Step:1] Set $w_{\ell m} := \frac{1}{|V|-1}$ for all $(\ell, m)$ such that $\ell \neq m$.
[Step:2] Continue [Step:3] and [Step:4] until convergence.
[Step:3] Solve (3) to get $\phi^{(\infty)}(i, j)$ for $(i, j) \in E^U$.
[Step:4] Find $W$ that maximizes $L(W)$, by using (4).
```

Figure 2. A batch transduction algorithm

```
Algorithm: Sequential
[Step:1] Set $w_{\ell m} := \frac{1}{|V|-1}$ for all $(\ell, m)$ such that $\ell \neq m$.
[Step:2] Solve (3) to get $\phi^{(\infty)}(i, j)$ for $(i, j) \in E^U$.
[Step:3] Continue [Step:4] until convergence.
[Step:4] Sample $(i, j)$ at uniformly random
  [Step:4-a] Update $\phi^{(\infty)}(i, j)$ by using (5) if $(i, j) \in E^U$,
  [Step:4-b] Update $w_{\ell m}$ for $\{(\ell, m) | \ell \in \{1, 2, \ldots, |V|\}, m \in \{i, j\}\}$ by using (6) if $(i, j) \in E^L$.
```

Figure 3. A sequential transduction algorithm

paths$^{(l)}_{i,j}$ is the number of paths of length l from node i to node j. It is essentially identical to the diffusion kernel used in kernel methods to define the similarity between two nodes on a graph. In our experiment, we set $\beta = 0.05$.

- Preferential attachment [17, 4]

  $$\text{preferential} := |\Gamma(i)| \cdot |\Gamma(j)|$$

  Preferential attachment is based on a different idea from those of the above variants of common neighbors metric. It originates with a generative model of scale free networks, where nodes with many neighbors are likely to obtain new neighbors.

5.2 Experimental setting

We used two biological network datasets in our experiments. One is a medium sized metabolic network dataset, and the other is a larger protein-protein interaction network dataset.

- Metabolic network
  The first dataset [27] contains metabolic pathways of the yeast S. Cerevisiae in KEGG/PATHWAY database [11]. Figure 1 shows an overview of this network. In this network, proteins are represented as nodes, and an edge indicates that the two proteins are enzymes that catalyze successive reactions between them.

  The number of nodes in the network is 618, and the number of links is 2782. Therefore the number of data (i.e. node pairs) to be classified is 190653, and the ratio of positive and negative data is $0.015 : 1$.

- Protein-protein interaction network
  The second dataset is a protein-protein interaction network dataset constructed by von Mering et al. [25]. We followed Tsuda and Noble [24], and used the medium confidence network, containing 2617 nodes and 11855 links. Therefore the number of data (i.e. node pairs) to be classified is 3423036, and the ratio of positive and negative data is $0.003 : 1$.

In both datasets, the number of edges are very small as compared to the number of node pairs, so the ratio of positive and negative data is highly skewed. Therefore, in [Step:4] in the learning algorithm of Figure 3, we use weighted sampling using the data distribution, that is, the positive data are sampled with probability proportional to the ratio of the number of negative data to the total number of data (and similarly for the negative data).

We used 66% of the data as training data and the rest as test data, and evaluated the performance by 3-fold cross validation. The performance of competing methods is compared using precision-recall curves since the dataset are highly skewed, with less than 2% of the edge labels being positive.

5.3 Experimental Results

First, we will investigate the relative performance of the various methods we consider. Figure 4 exhibits the precision-recall curves for the metabolic network data, and Figure 4 shows those for the protein-protein interaction network data. Table 1 summarizes these results, in terms of the break-even points of precision and recall by the respective methods.

Overall, the proposed method achieves better performance than all other methods, particularly with the metabolic network data. With the protein-protein network data, the improvement is not as dramatic, especially in terms of the break-even points.

In link prediction, however, it is generally considered important to achieve high precision in the low recall area (i.e. the left-half of the precision-recall curve). This is because, in actual applications, link prediction is often used to recommend a small number of promising pairs from among a large set of candidates, e.g. recommending new friends in social network services or potential protein-protein interactions that have not been found experimentally. From this perspective, we conclude, the proposed method enjoys a significantly higher performance.

Next, we see whether the transductive formulation results in significant improvement in predictive performance. We can obtain a non-transduction version of our link prediction method by simply setting $\phi(i,j)$ for $(i,j) \in E^L$ to be identically 0. Figure 6 shows the precision recall curves of the proposed method, with and without transduction, for the metabolic network data. We observe that adoption of transductive formulation does boost the predictive performances for these datasets.

Finally, we examine the relationship among the competing methods by evaluating the similarity of predictions output by them. In particular, we compare them by using Spearman rank correlation [9] on the positive test data. Spearman rank correlation is a measure of how two rankings of a given set are similar to each other. The similarity is the highest when it is 1, and the lowest when it is 0. Table 2 shows the Spearman rank correlations for all pairs of the methods we consider, for the metabolic network data. Also, Figure 7 visualizes them in two dimensions by multidimensional scaling [5]. Note that they are essentially comparisons for only the positive test data, since the datasets are highly skewed. Therefore the prediction performance can differ significantly even when the two rankings are highly correlated. (It is the ranking on the positive test data that matters in attaining high predictive performance, of which there are few.) Now, let us examine the results. We can observe that the predictions made by common neighbor, Jaccard's coefficient, and Adamic/Adar are very similar to one another, which is actually implied by their definitions. Since Katz$_\beta$ can be regarded as an extension of these metrics, it is no surprise that it also has high correlations with them, but at the same time, it is also similar to preferential attachment.

It is interesting to observe that the mean correlation for the proposed method is the lowest among all methods, implying that the proposed method is rather different from all the other methods, in the way it predicts. It is consistent with the fact that our model is based on an evolution model with different characteristics from the models that the other metrics/methods are based upon. This means that adding the proposed method to the pool of existing methods, for example, when using their predictions as part of input features to a subsequent classifier, would be beneficial.

	metabolic	protein-protein
common	21.1%	37.9%
Jaccard's	30.2%	47.9%
Adamic/Adar	34.9%	50.3%
preferential	9.2%	25.4%
Katz$_{0.05}$	32.8%	27.6%
proposed	61.2%	52.6%

Table 1. Summary of results for both dataset measured by break-even point of precision and recall.

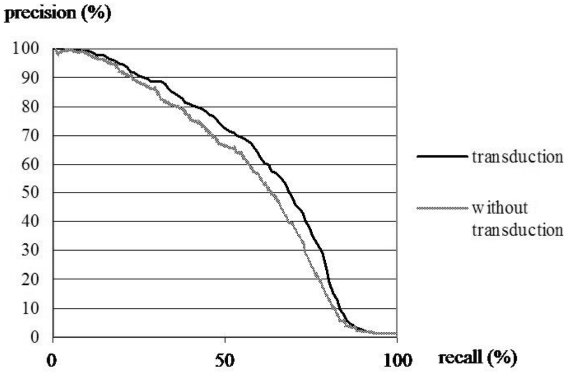

Figure 6. Comparison of Precision-Recall curves of transductive inference and non-transductive inference for metabolic network. The break-even points are 61.2% with transduction, and 58.0% without transduction, respectively.

6 Related work

As we pointed out in Introduction, link prediction is naturally cast as a classification/ranking problem for node pairs, and two types of information, node features and topological features, are used for this task.

Topological features are often derived from generative models of network structure [17, 3, 15, 1, 12, 17, 4], and our model can be interpreted as a parameterized version of the "node copying model" due to Kleinberg et al. [14, 13], To the best of our knowledge, there been no probabilis-

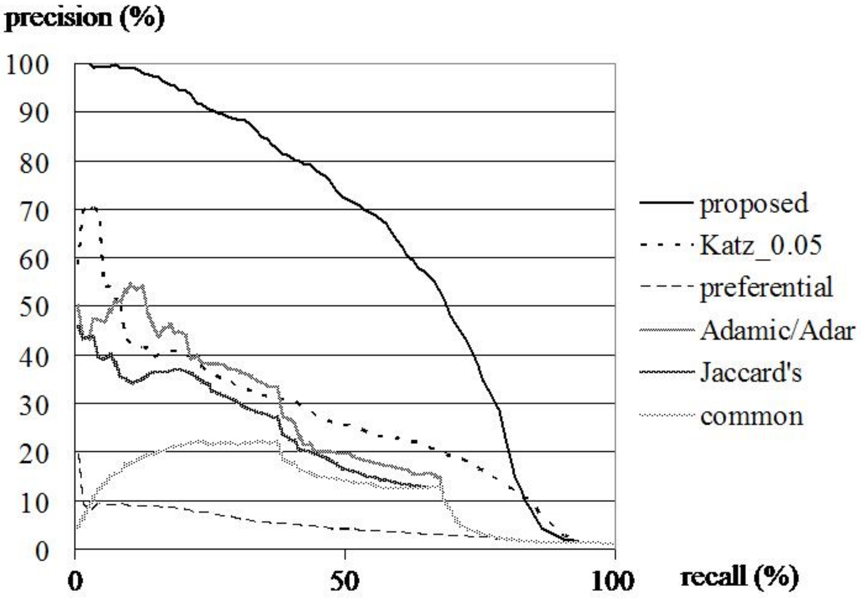

Figure 4. Precision-Recall curve for metabolic network with 66% training data.

	c	J	A	p	K	proposed
common	1	0.92	0.94	0.31	0.61	0.20
Jaccard's	0.92	1	0.97	0.53	0.75	0.35
Adamic/Adar	0.94	0.97	1	0.49	0.70	0.31
preferential	0.31	0.53	0.49	1	0.84	0.69
$\text{Katz}_{0.05}$	0.61	0.75	0.70	0.84	1	0.70
proposed	0.20	0.35	0.31	0.69	0.70	1
mean	0.80	0.91	0.88	0.77	0.92	0.65

Table 2. Spearman rank correlations among rankings by various methods for positive test data in the metabolic network dataset. Note that the Spearman rank correlation is symmetric. Columns indicated by c, J, A, p, and K denote common neighbors, Jaccard's coefficient, Adamic/Adar, preferential attachment, and $\text{Katz}_{0.05}$, respectively.

tic models with tunable parameters derived from a network evolution model, nor have there been any topological metrics defined based on a node copying model. We also point out that our generative model differs from theirs, in that in our model the edge labels (either positive or negative) are copied, not the edges (positive labels only). Also, we note that our model does not consider node addition and deletion, since our interest is in link prediction. Another difference is that Kleinberg et al's model allows copying multiple edges at a time, our model copies one edge label at a time, where copy probabilities are parameterized and tunable in the model.

Another approach to link prediction that exists in the literature is that of applying supervised learning methods using node features as well as topological features. (See, for example, Hasan et al. [7] and O'Madadhain et al. [20].) We point out that their work differs from ours in that their learning algorithms do not learn parameters within the network model. In order to examine the merit of applying a supervised learning method on topological features, we conducted a preliminary experiment in which a linear predictor was applied on the topological features considered in our experiments to perform link prediction. We did not observe any accuracy improvement over the direct methods that base their predictions solely on the topological metrics.

There has also been some works that apply the framework of statistical relational learning to link prediction. For example, Popescul and Ungar [21] and Taskar et al. [23] have applied such approach, using both node features and topological features, including those we used in our experiments. It is worth noting that these are applications of a more general paradigm (of relational learning) to the link prediction problem, and are to be contrasted with approaches such as ours that develop tailored models and methods for link prediction per se.

Link prediction has strong resemblance to the problem

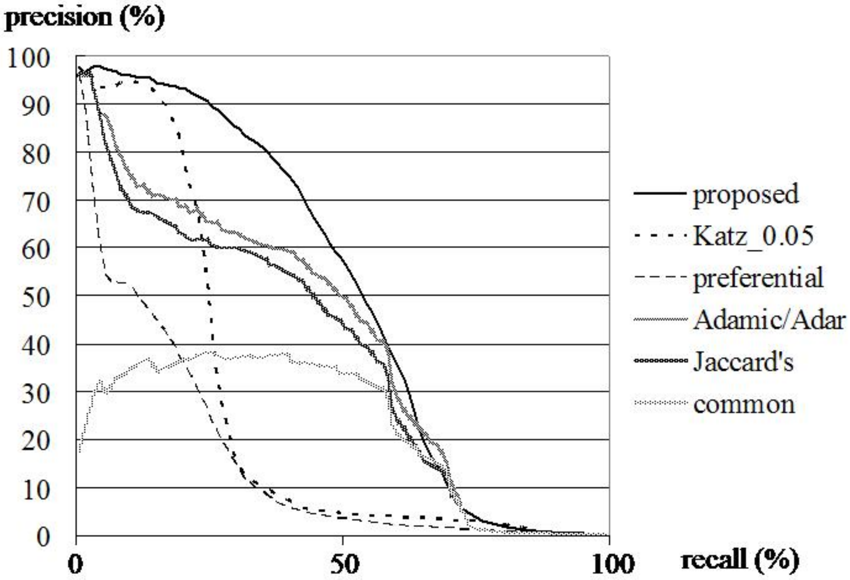

Figure 5. Precision-Recall curve for protein-to-protein interaction network with 66% training data.

of collaborative filtering, if we regard it as a matrix completion problem. For example, Huang et al. [28] applied the metrics we used in our experiments to collaborative filtering. On the other hand, application of collaborative filtering techniques to link prediction would also make sense. Indeed, an approach by Nakamura and Abe [16], which completes an user-item matrix by learning similarities among rows and columns, has some superficial resemblance to our approach and motivated our present work to some extent. Applications of other types of collaborative filtering techniques would be worth exploring.[1]

7 Conclusion and discussion

In this paper, we introduced a new approach to the problem of link prediction for network structured domains based on the topological features of network structure, not on the node features. We presented a probabilistic evolution model of network structure which models probabilistic flips of existence of edges depending on a “copy-and-paste” mechanism of edges. Based on this model, we proposed a transductive learning algorithm for link prediction based on an assumption of the stationarity of the network. Finally, we show some promising experimental results using real network data. We used biological network data in our experiments, and attained good performance. This is thought to be, in part, attributable to the fact that our network evolution model matches the characteristics of biological networks. Assessing the applicability of the proposed approach in other domains will be important in the future.

Our model can be easily generalized to the scenario in which there are more than two edge labels. Let Σ be a set of label types, and $\phi_a^{(t)}(i,j)$ be the edge label for each edge type $a \in \Sigma$. Then we can modify the model (2) as

$$\phi_a^{(t+1)}(i,j) = \frac{1}{|V|-1}\Big(\sum_{k\neq i,j} w_{kj}\phi_a^{(t)}(k,i) + w_{ki}\phi_a^{(t)}(k,j)\Big)$$
$$+\Big(1 - \frac{1}{|V|-1}\sum_{k\neq i,j} w_{kj} + w_{ki}\Big)\phi_a^{(t)}(i,j).$$

For example, this generalized model includes the case of directed graphs. There are numerous domains in the real world to which directed networks can be applied. In the future, we wish to apply our approach to such networks and compare its performance with the directed versions of the metrics considered in this paper.

Acknowledgements

Hisashi Kashima thanks Prof. Tatsuya Akutsu and Dr. Yoshihiro Yamanishi of Kyoto University Bioinformatics Center, and Dr. Koji Tsuda and Dr. Hiroto Saigo of Max

[1] Although we did not show the results in this paper, the matrix factorization approach [22] performed poorly on link prediction tasks in our preliminary experiments.

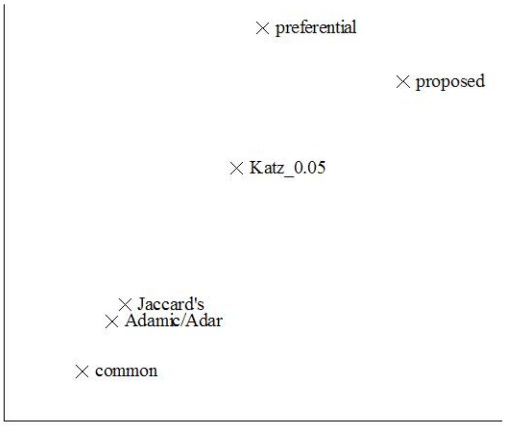

Figure 7. Visualization of the Spearman correlation matrix of Table 2 in two dimensions by multi-dimensional scaling.

Planck Institute for Biological Cybernetics for valuable discussions. Authors thank Yuta Tsuboi of IBM Tokyo Research Laboratory for helping with conducting the experiments.

References

[1] L. A. Adamic and E. Adar. Friends and neighbors on the Web. *Social Networks*, 25(2):211–230, 2003.

[2] R. Albert, A. Barabási, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–187, 1999.

[3] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[4] A. L. Barabási, J. Jeong, Z. Néda., E. Ravasz, A. Shubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311(3-4):590–614, 2002.

[5] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling, 2nd ed.* Chapman and Hall, 2004.

[6] L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explorations*, 7(2):3–12, 2005.

[7] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005)*, 2005.

[8] D. Helmbold, R. Schapire, Y. Singer, and M. Warmuth. A comparison of new and old algorithms for a mixture estimation problem. In *Proceedings of the Eighth Annual Workshop on Computational Learning Theory (COLT)*, pages 69–78, 1995.

[9] R. V. Hogg and Craig. *Introduction to Mathematical Statistics, 5th ed.* Macmillan, 1995.

[10] T. Ide and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the Tenth ACM SIGKDD Conference (KDD)*, 2004.

[11] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resources for deciphering the genome. *Nucleic Acids Research*, 32:D277–D280, 2004.

[12] L. Katz. A new status index derived from sociometric analysis. *Psycometrika*, 18(1):39–43, 1953.

[13] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–17, 1999.

[14] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[15] D. Liben-Nowelly and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM)*, pages 556–559, 2004.

[16] A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, pages 395–403, 1998.

[17] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review Letters E*, 64(025102), 2001.

[18] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[19] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

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

[21] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, 2003.

[22] N. Srebro. *Learning with Matrix Factorization*. PhD thesis, Massachusetts Institute of Technology, 2004.

[23] B. Taskar, M. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing System*, 2003.

[24] K. Tsuda and W. S. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20(Suppl. 1):i326–i333, 2004.

[25] C. von Mering, R. Krause, B. Snel, M. Cornell, S. Olivier, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403, 2002.

[26] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[27] Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics*, 21:i468–i477, 2005.

[28] H. C. Zan Huang, Xin Li. Link prediction approach to collaborative filtering. In *Proceedings of the Fifth ACM/IEEE-CS joint conference on Digital libraries (JCDL)*, 2005.