INFORMS Journal on Computing
Vol. 27, No. 2, Spring 2015, pp. 249–267
ISSN 1091-9856 (print) | ISSN 1526-5528 (online)

http://dx.doi.org/10.1287/ijoc.2014.0624

A Network Structural Approach to the Link Prediction Problem

Chungmok Lee
School of Industrial and Management Engineering, Hankuk University of Foreign Studies,
Yongin 449-791, Republic of Korea, chungmok@hufs.ac.kr

Minh Pham
SAMSI and Duke University, Durham, North Carolina 27707, ptuanminh@gmail.com

Myong K. Jeong
RUTCOR, Rutgers University, Piscataway, New Jersey 08854, mjeong@rci.rutgers.edu

Dohyun Kim
Department of Industrial and Management Engineering, Myongji University, Yongin 449-728, Republic of Korea,
ftgog@mju.ac.kr

Dennis K. J. Lin
Department of Statistics, Eberly College of Sciences, Pennsylvania State University, University Park, Pennsylvania 16802,
dennislin@psu.edu

Wanpracha Art Chavalitwongse
Departments of Industrial and Systems Engineering and Radiology, University of Washington, Seattle, Washington 98195,
artchao@uw.edu

The link prediction problem is an emerging real-life social network problem in which data mining techniques have played a critical role. It arises in many practical applications such as recommender systems, information retrieval, and marketing analysis of social networks. We propose a new mathematical programming approach for predicting a future network using estimated node degree distribution identified from historical data. The link prediction problem is formulated as an integer programming problem that maximizes the sum of link scores (probabilities) with respect to the estimated node degree distribution. The performance of the proposed framework is tested on real-life social networks, and the computational results show that the proposed approach can improve the performance of previously published link prediction methods.

Keywords: analysis of algorithms; data mining; link prediction; optimization
History: Accepted by Alexander Tuzhilin, (former) Area Editor for Knowledge and Data Management; received September 2011; revised October 2012, September 2013, March 2014; accepted August 2014.

1. Introduction

The link prediction problem is an emerging data mining problem the goal of which is to predict the existence of a link between every node pair in the network based on the past network topology (Lu and Zhou 2011). Many real-world complex systems, such as those arising from biological and social interactions, can be described by *network* representation. In such networks, each node represents a participator in the system, and the links (or edges) represent the existence of connections (or sufficient similarities) between nodes. Generally speaking, the objective of many data mining problems is to identify (or predict) hidden structure(s) of the network(s), if any, based on available knowledge of the system(s). In the same fashion, the link prediction is a data mining problem that appears in many research areas. For example, the aim of information retrieval is to classify unidentified documents by predicting the relationships between words and document classes, where each node denotes a word or a document class (Salton 1989, Manning et al. 2008). The analysis of biological interactions is another example of a scientific field in which the link prediction problem is clearly relevant primarily because of the high experimental costs for large biological networks. Bader et al. (2003) modeled the problem of predicting the biological relevance of protein-protein interactions as a link prediction problem and developed a logistic regression approach using the statistical and topological properties of the protein network. An overview of data mining techniques in the context of the protein-protein interaction networks was presented by Mamitsuka (2012). Goldberg and Roth (2003) exploited the local cohesiveness property of the protein network to predict the missing links of the (possibly error-prone) experimentally derived graphs. The recommender system is another important application of the link prediction problem. Huang et al. (2005) adapted a number of graph theoretic measures between the users and the items to obtain a recommendation of books. In this case, the system is represented in a user-item bipartite network, and a link between a user and a book denotes the preference between them.

The link prediction problem can be applied in evolving networks also. For instance, how the structure of Internet topology evolves over time has been an important question in computer science and social science (Medina et al. 2000, Zhou and Mondragón 2004). Since the topology of the Internet can be represented as a network of connections (links), the link prediction can be used to analyze and/or predict the future shape of the Internet. Recently, large-scale social networks like Facebook and Twitter have emerged, and predicting the future connections (e.g., friend or follower) of the users will be practically useful. Hoff (2009) introduced a latent factor model to incorporate the high-order correlation effects of the international conflict networks for predicting the missing links. Predicting the prospective links in the co-authorship network was investigated by Al Hasan et al. (2006). They treated the link prediction as a supervised learning problem and adapted the feature selection procedure to identify the most important features between the researchers. Juszczyszyn et al. (2012) investigated a predictive model of structural changes of subgraphs of a university email network by using Markov chain.

The simplest (and arguably most effective) algorithms for solving the link prediction problem are the so-called *scoring methods*. In scoring methods, a number of scoring functions that measure the similarity (or proximity) between the individuals of the network are defined. For each link of the network that needs to be predicted, the scoring function assigns a certain score to the link, and the score itself (often informally) represents the probability of the existence of the link. The scoring functions can be defined in various ways, with each method designed to reflect a specific aspect of the network topology, such as the number of neighbors, the distance, and/or the clusters. Once the link scores are calculated, the prediction can be made by sorting the link scores in decreasing order and choosing a predefined number of links with top scores. Liben-Nowell and Kleinberg (2007) compared the prediction performances of many scoring methods on the co-authorship network.

Because of its simplicity, the scoring method is quite flexible and can be incorporated with more sophisticated methods. When each node of the network has some attributes, the score can be calculated by using a supervised learning framework such as regression. In fact, any scoring function may be treated as an attribute or feature of the node and/or the link. Al Hasan et al. (2006) compared different classes of supervised learning algorithms for the link prediction on bibliographic data sets. In addition, when the networks have some probabilistic inferences, probabilistic approaches can be considered. The probability relational models find hidden parameters that can best explain the observed data. The probability of the existence of a link is obtained by the conditional probability on the estimated parameters. Many variations of the probability relational model were proposed in various alternative forms such as a binary tree (Clauset et al. 2008, Park et al. 2010), a generative Bayesian model (Herlau et al. 2012), and local fitness function (Lancichinetti et al. 2009).

Almost all link algorithms in the literature can essentially be considered as an estimation method of the existence probability of each link—i.e., they can be considered as local information. In other words, one calculates the score (e.g., probability or similarity) of each single link, and the only criterion of prediction is the score. In this context, those link prediction algorithms may be seen as *greedy* algorithms. Given that (i) the algorithms completely depend on a limited amount of data that have already been observed, and (ii) the prediction is made in a greedy manner, there may be overfitting issues. In many data mining frameworks, the overfitting issues can often be remedied by introducing some forms of *regularization*. Regularization methods are based on a priori knowledge of the problem, such as the widely used parsimonious assumption. In other words, the generalization performance can be improved by regulating (or guiding) the prediction phase through the use of a priori knowledge of the network.

In this study, we propose a novel link prediction framework that regulates the network by means of node degree distribution to characterize the network. Recently, research on social networks has revealed the existence of the *power law* of the node degree distribution (Barabási and Albert 1999, Boginski et al. 2006, Jeong et al. 2000, Kim et al. 2002). Here, we present a mathematical programming approach that maximizes the sum of the scores of the links predicted while concomitantly considering the node degree distribution so that the prediction phase will not be too greedy. The proposed framework can be integrated smoothly with any existing scoring method and can be seen as generally resembling the Bayesian framework; i.e., our prior belief of the node degree distribution being a specific characteristic of the network is exploited to regulate the link prediction.

This paper is organized as follows: In the §2, we briefly review the existing link prediction algorithms. In §3, the network structural approach for the link prediction problem is introduced. The mathematical formulation and its solution algorithm are developed in §4, and the computational study for the real-life networks is reported in §5. Our conclusions are given in §6.

2. Background

The scoring method is arguably the most widely used link prediction algorithm, as it can be quite effective. The essential element of the scoring method is the definition of the scoring function, which assigns score values to every link to be predicted. The basis of the score function is that the higher the score, the more likely it is that the link will appear in the future network. A common principle of all scoring methods is that the absolute magnitudes of the scores are not important and that only the ordering of the link scores matters. After assigning a score to each link to be predicted, the prediction is made by taking n^* links with the highest scores. Various approaches can be used to define the score function, and they can be divided into two major categories: (i) static representation of networks (scoring algorithms based on a static graph) and (ii) dynamic representation (exploiting the temporal aspects of the networks using time-series analysis). Most of the link prediction algorithms developed to date are based on the static representation of networks, which can be obtained by aggregating the past network data. After building a static graph from past observations, we use a number of graph theoretic measures to assign scores to the links to be predicted. The underlying motivation of this approach is that the topological properties of any link in past networks are likely to be complied in the future network. Huang and Lin (2009) recently proposed a time-series (TS) analysis approach for the link prediction problem. Based on their results, they concluded that the best performance was shown by the TS analysis approach combined with static scoring methods. In particular, they showed that this approach and the various static scoring methods have the potential to complement each other, as they represent two different aspects of the network. da Silva Soares and Bastos Cavalcante Prudencio (2012) considered a similar TS approach in conjunction with unsupervised and supervised link prediction models, and showed satisfactory results for the co-authorship networks. In the following, we briefly review some of the most popular scoring algorithms. All but the TS analysis scoring method are categorized as static scoring methods.

Adamic/Addr (*ADA*): Adamic and Adar (2003) proposed a scoring method for measuring the similarity between two websites i and j, which is defined as

$$\sum_{\substack{k:\,\text{feature shared}\\ \text{by } i \text{ and } j}} \frac{1}{\log\, \mathit{frequency}(k)}.$$

In the link prediction setting, the features of site i can be the set of adjacent nodes of site i. Let $\Gamma(i)$ denote the set of adjacent (neighbor) nodes of site i on the observed network. Let $|A|$ denote cardinality of a set A. Then, the ADA score s_{ij}^{ADA} for link $\{i, j\}$ can be given as:

$$s_{ij}^{\text{ADA}} := \sum_{k\in\Gamma(i)\cap\Gamma(j)} \frac{1}{\log|\Gamma(k)|}. \tag{1}$$

Katz (*KZ*): The Katz measure is obtained by summing the number of paths of length l between two nodes i and j for all l (Katz 1953). In the summation, the number of paths of length l is multiplied by damping parameter β^l to prevent divergence. Let M denote the adjacency matrix of the observed graph. It is well known that the number of paths with length l between two nodes i and j is the (i, j) element of M^l. Therefore, the summation over all l gives the following score:

$$s_{ij}^{\text{KZ}} := \left[\sum_{l=1,\ldots,\infty} \beta^l M^l\right]_{ij} = [(I-\beta M)^{-1} - I]_{ij}. \tag{2}$$

Preferential Attachment (*PA*): This measure was originally introduced to explain the preferential attachment phenomena often observed in real-life large networks (Barabási and Albert 1999). Some studies on social networks subsequently showed that the probability of interaction between two nodes is correlated with the product of the degrees of two nodes (Barabási et al. 2002; Newman 2001a, b). Therefore, the score for link $\{i, j\}$ is given as:

$$s_{ij}^{\text{PA}} := |\Gamma(i)|\cdot|\Gamma(j)|. \tag{3}$$

Time-Series (*TS*) *Analysis*: The score methods mentioned do not exploit the temporal aspect of networks. Huang and Lin (2009) proposed a TS approach in which the basic premise is to use the TS frequency information of link occurrence obtained by fitting the ARIMA model (Box et al. 1970). For each link, these authors considered the link frequency TS data from past networks and applied the ARIMA model to predict the probability of the future appearance of the link. The computational results showed that the combined approach of the TS and static methods outperformed the previous methods using static information only. Let $\hat{x}_{ij}$ and sd_{ij} denote the predicted link frequency and the standard deviation of the prediction for the link $\{i, j\}$ in the future network by the ARIMA model, respectively. Then, the score for the link is given as:

$$s_{ij}^{\text{TS}} := \Pr(\hat{x}_{ij} > \theta), \tag{4}$$

where θ is a given constant and the probability is calculated from a normal distribution $\mathcal{N}(\hat{x}_{ij}, sd_{ij}^2)$.

Hierarchical Random Graph Model (*HRM*): This method, first proposed by Clauset et al. (2008), exploits the hierarchical structure of social networks. In this model, the hierarchical structure of networks is represented by a dendrogram where closely related pairs of individuals have lower common ancestors. Each inner node r in the dendrogram has some probability p_r that any pair of individuals in left and right subtrees will have any connections. For a given dendrogram, the probability p_r can be calculated easily by maximizing the likelihood of dendrogram that generates the observed network, and for any individual pair i and j the probability p_{ij} that they are connected by a link is $p_{ij} = p_r$, where r is the lowest common ancestor in the dendrogram. They used the Markov chain Monte Carlo algorithm to sample many dendrograms with good likelihood measures. After sampling, the probability of a link between i and j is given as the average of p_{ij} for all sampled dendrograms. This is equivalent to assigning the score s_{ij}^{HRM} for the link $\{i, j\}$ as follows:

$$s_{ij}^{\mathrm{HRM}} := \frac{1}{|D|} \sum_{d \in D} p_{ij}^{d}, \tag{5}$$

where D is a set of sampled dendrograms.

Any scoring algorithm can be used by itself or combined with other scoring algorithms. Moreover, the scoring method may serve as a starting point for more sophisticated link prediction algorithms, such as the probabilistic classification method (Hoff et al. 2002, Hoff 2009). We refer the reader to the survey paper by Newman (2001a) for more detailed descriptions of the scoring methods.

3. A Network Structural Approach

In this section, we propose a new network structural approach that uses the information available from the node degree distribution to predict the links in the future networks. The basic concept is to predict a future network topology based on the scores obtained from a number of scoring algorithms while concomitantly restricting the predicted network to a certain global structure of the present network: the node degree distribution.

The degree of a node represents the number of incident links to the node in the network. In many social networks, the node degree is a basic quantification of how actively the node is interconnected to other nodes. Let $\mathcal{P}(d)$ denote the probability (when it is divided by the total number of the nodes) of any node with node degree d in the network. Then $\mathcal{P}(d)$ can be estimated based on the frequency of nodes having node degree d in the network. Since the seminal work by Barabási and Albert (1999) was published, it has turned out that nearly every real-life network has a specific form of node degree distribution, e.g., the power law degree distribution (Newman 2003).

In fact, the power law-like degree distribution is a somewhat unexpected result, since in the random graph model developed by Erdos and Renyi (1959), the degree distribution is expected to show a more centralized distribution. The easiest way to identify the existence of the power law degree distribution in the network is by plotting the degree distribution in a log-log scale. In a log-log scaled plot, the power law distribution takes on the appearance of a straight line with a negative slope, which implies that the degree distribution function has a form resembling

$$\mathcal{P}(d) \propto d^{-\alpha}, \tag{6}$$

where α is a constant that varies with the network type. One notable aspect of the power law-like degree distribution is that it makes the network scale-free—any network that shows the power law degree distribution is often referred to as the scale-free network. The implications of a network being scale-free are (i) the power law degree distribution holds regardless of the size of the network and (ii) the power law degree distribution property is complied with even if the network is growing (or shrinking). It can therefore be said that if there is some power law-like degree distribution in the past network, the future network can be expected to follow the same distribution.

We can view the degree distribution as some sort of global characteristics of the network. We naturally expect that the graph to be predicted also follows the degree distribution observed in the past networks. To achieve this, a link prediction method should explicitly take into account a specific node degree distribution of observed networks in making predictions. There are very limited studies explicitly addressing degree distribution in the link prediction settings, however, some rare exceptions are found in community detection literature. Jiang and Tuzhilin (2009, p. 319) mentioned in their paper:

> As with many natural phenomenon that has a power law distribution, our result suggest that the decline rate in terms of segment counts per segment size, starting from the peak of the segment size distribution, would also follow a power law distribution for the optimal solution. However, formal analysis is required to prove this conjecture.

They suspected that the reason for the superior performance of their algorithm is that their algorithm happens to produce a power law distribution, which is not intended. Karrer and Newman (2011) proposed a degree-corrected stochastic block model to incorporate *degree heterogeneity* of communities. They found that a standard stochastic block model without consideration of degree distribution is likely to miss important structure of real-life networks, so it can give radically incorrect answers. Roughly speaking, the degree-corrected stochastic block model essentially minimizes a Kullback-Leibler divergence between p_K and p_{degree}, where p_K is the probability distribution of given block model and p_{degree} is the probability distribution produced by the preferential attachment model that consequently results in a power law degree distribution. Therefore, their model finds a block model whose degree distribution of nodes in a community is most similar to the power law-like degree distribution. They showed that incorporation of degree distribution property in the stochastic block model could perform much better in detecting real-life community structure. Shen et al. (2011), Chaudhuri et al. (2012), and Newman (2012) also indicate that incorporating global structures such as degree distribution can improve the prediction performance significantly. In this paper, we share the motivation of Karrer and Newman's approach, and in the following sections propose a somewhat different approach: we propose an optimization model that ensures the predicted graph does indeed follow the specified node degree distribution.

Most of the scoring algorithms reviewed in §2 were designed to assess certain *local* structural characteristics of the observed networks. It is commonly recognized that a particular scoring method performs well for certain networks but fails to provide a good prediction for other networks. This shortcoming may be overcome by carefully combining several scoring methods. Theoretical results on just how to combine different scoring functions are, however, limited. The criticism of the scoring methods can be summed up simply: they are too *microscopic*. Generally speaking, the scoring functions are defined on a single link; this rather restrictive definition means that it is difficult to assess the network as a whole. For example, the similarity (or score) between two links may not be directly observable from the historical occurrence of each link and may depend on some complicated global (or collective) structural aspects of the networks. The lack of a *macroscopic* consideration of the scoring methods becomes more evident when the actual prediction is made. Since we only choose a certain number of links with the highest scores, the possible correlations between links may be ignored. For example, suppose that two links having similar scores to each other are so extremely negatively correlated that only one of them can appear at a time. The scoring methods cannot reflect this kind of complex constraint (or network characteristic) since the scores of the links are the only criterion used by the algorithms to make the prediction.

Statistical techniques, such as correlation analysis, can be used to extract information on the collective structure of the links in the network. In this approach, we see that each participant in the network has a collection of attributes. Based on these attributes, the correlation analysis can be performed using information compiled by observing the networks. There are many variations of the procedure used for handling (or defining) the correlation and the similarity between the links (see Kolaczyk 2009, §7.3 for details). Despite the clear relevance of the correlation analysis to the link prediction application, few studies have focused on this line of approach. There are a number of reasons for this lack. First, in many cases, there is not enough information to induce meaningful statistical inference. Generally speaking, the statistical analysis is based on the implicit assumption of random and unbiased repetition of outcomes, but this assumption is not always valid. An excessively fine-grained statistical analysis on an insufficient body of data can lead to a critically misleading interpretation. Second, a statistical analysis can be extremely time consuming when the network under study is very large, as is the case in many real-world social network cases. The computational burden becomes even larger when we consider the time-dependent correlations (TS approaches) as well as the topological-dependent correlations of the networks. Third, in some cases, the correlation merely provides some information on the results (or outcomes) of the networks and does not extend our knowledge on what causes the results (correlation does not imply causation). Therefore, the correlation structure may represent the less essential characteristics of the networks since the correlation may vary over time. As an example, consider a cooperative network between employees of a company. The degree of interactions between any two employees can change because of promotion, transfer, or company reorganization. We believe that the scoring algorithms are too coarse (since they do not consider the global characteristics) and the statistical analysis approaches are too fine (since they may suffer from some overfitting issues). The use of node degree information in link prediction can be seen as falling in between these two approaches.

It may be argued whether complying with the degree distributions is always desirable. For example, some link recommendation applications seek links connecting different community groups that would never be connected otherwise. The motivation of such application is to stimulate interactions between the communities so that the network could evolve actively in a somewhat unexpected way. In this setting, complying with the degree distribution is not desirable because the major interest is to manipulate the network structure rather than preserve it. This observation enables us to classify the link prediction-related applications into two broad categories: (i) predicting (or detecting) future (or missing) links of the networks and (ii) making (or recommending) links to alter the future network structure. Here, we only focus on the prediction problems in which no modification or altering of network is allowed. In this case, complying with the node degree distribution does not mean we restrain the degree distribution of real-world networks because we have no control methods for the real world. It simply means the degree distribution of the prediction solutions complies with the expectation that a solution with similar statistical properties to the real-world networks would be better in simulating its real-world counterpart. The prediction problems can be classified further into two subproblems: (i) construction of the future networks by predicting the existence of a link between each node pair; and (ii) identification of additional links to be augmented in the future to an existing network. Here, we consider both subproblems.

4. Solution Methodology

Let $G_t(V, E_t)$ denote the undirected graph of the network at time t, where $V := \{1, \dots, N\}$ is the set of nodes and E_t is the set of observed links at time t. The link prediction problem is to predict a set of links E_T at time T based on previous knowledge of $E_1, \dots, E_{T-1}$. In this paper, we assume that the set of nodes remains the same for all values of the time index $t = 1, \dots, T$, which means that the size of the network remains the same in terms of the number of nodes. For growing (or shrinking) networks, we can add a number of dummy nodes to ensure that the graphs have the same number of nodes over all time periods. We basically consider the undirected graph, although the proposed approach can be readily extended to the directed case.

For each (unordered) pair of nodes $i \in V$ and $j \in V$, let $s(i, j)$ (or s_e) denote the score of the link (i, j) (or e). The score $s(i, j)$ is computed using various link scoring methods. Then, the sets of predicted links are obtained in all conventional link prediction algorithms by applying a threshold value s^*, which is equivalent to taking the top n^* scored links after ordering the links, where n^* determines the number of links of the predicted network. For this reason, we call this kind of algorithm the *simple ordering* (SO) algorithm. Consider an $N \times N$ matrix S whose element s_{ij} is given as some specific score $s(i, j)$. We call S a *score matrix*. The SO algorithms can be stated as follows:

$$(\mathrm{P}_{\mathrm{SO}}) \quad \max_{x \in \{0,1\}^{|E|}} \left\{ \sum_{e \in E} s_e x_e \,\middle|\, \sum_{e \in E} x_e \le n^* \right\}, \tag{7}$$

where S is the score matrix calculated from our knowledge of the previous networks $G_1, \dots, G_{T-1}$. The set E is the set of link candidates to be predicted; usually $E := \{\{i, j\} \mid i \neq j,\ i \in V,\ j \in V\}$. The decision variable x_e is 1 if link e is predicted, and 0 otherwise. In fact, problem (7) can be solved easily by sorting all elements of the score matrix and choosing the top n^* links.

We now assume that the estimated probability distribution of node degrees $\hat{\mathscr{P}}(d)$ for the network G_t for period $t = T$ is obtained from the past networks $G_1, \dots, G_{T-1}$. For some nonnegative integer vector $b \in \mathbb{Z}_+^N$, we denote $b \sim \hat{\mathscr{P}}$ if the value distribution of b approximately follows the node degree distribution $\hat{\mathscr{P}}$ (where b_i is the ith component of b, representing the node degree value of node $i \in N$). For some nonnegative integer vector $\hat{b} \sim \hat{\mathscr{P}}$, let B denote a set of all element-wise permutations of $\hat{b}$; i.e., $B := \{b \in \mathbb{Z}_+^N \mid b = P\hat{b}$, for some permutation matrix $P\}$. The link prediction problem with the aim of preserving the node degree distribution at the future network can then be stated as follows:

$$(\mathrm{P}_{\mathrm{DD}}) \quad \max_{b \in B} \max_{x \in \{0,1\}^{|E|}} \left\{ \sum_{e \in E} s_e x_e \,\middle|\, \sum_{e \in \sigma_i} x_e \le b_i, \forall i = 1, \dots, N \right\}, \tag{8}$$

where σ_i is the set of the links adjacent to node i; i.e., $\sigma_i := \{\{i, j\} \in E \mid j \neq i,\ j \in V\}$. We call problem (8) the degree distributional approach (DD). In this approach, we assume that the same node degree distribution property of the previous networks $G_1, \dots, G_{T-1}$ also exists in the newly predicted (future) networks G_T. The objective of this formulation is to find a network that maximizes the sum of link scores while respecting the node degree distribution. Note that we do not specify the node degree of any node i here. We only restrict the *distribution* of the node degrees. Also note that the inner maximization problem of $(\mathrm{P}_{\mathrm{DD}})$ is a maximum-weight b-matching problem that has a polynomial time algorithm (Cook and Pulleyblank 1987, Anstee 1987). The problem $(\mathrm{P}_{\mathrm{DD}})$ is, however, NP-hard, as shown in the following.

4.1. Computational Complexity of $(\mathrm{P}_{\mathrm{DD}})$

Let $F(s, b) := \max_{x \in \{0,1\}^{|E|}} \{\sum_{e \in E} s_e x_e \mid \sum_{e \in \sigma_i} x_e \le b_i,\ \forall i = 1, \dots, N\}$; then we can formally define a decision version of the problem $(\mathrm{P}_{\mathrm{DD}})$ as follows. Without loss of generality we assume all data are integer.

Problem 1 (Maximum-Weight b-Matching Over Permutation Group).
Instance: Undirected graph $G(V, E)$, nonnegative integer vectors $\hat{b} \in \mathbb{Z}_+^{|V|}$ and $s \in \mathbb{Z}_+^{|E|}$, and positive integer $L \le \sum_{e \in E} s_e$.
Question: Determine if $\max_{b \in B} F(s, b) \ge L$ (i.e., is there a permutation matrix P such that $F(s, P\hat{b}) \ge L$?).

Theorem 1. *Problem* 1 *is NP-complete.*

Proof. Reduction from a satisfiability problem (SAT). See online supplement (available as supplemental material at http://dx.doi.org/10.1287/ijoc.2014.0624) for details. □

4.2. Approximating Scheme

Since the problem (P_{DD}) is NP-hard, solving the problem directly is not practical especially for large networks. Therefore, we propose the following approximating scheme.

For a given number K, we divide the range of the node degrees into K intervals, as illustrated in Figure 1. Let a_k for all $k = 1, \ldots, K+1$ denote the dividing points. The number of nodes belonging to the interval k can then be calculated by the degree distribution function. Let g_k denote the number of nodes having the node degrees belonging to interval k. This results in

$$g_k := \left[N \times \int_{a_k}^{a_{k+1}} \hat{\mathscr{P}}(z)\, dz \right], \tag{9}$$

where $[\cdot]$ is a function that returns the nearest integer. We now introduce binary variables y_i^k, whose value is 1 if node i has node degree a_k, and 0 otherwise. The following problem is then obtained as:

$$(\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}}) \quad \text{maximize} \quad \left\{ \sum_{e \in E} s_e x_e - D \sum_{i \in V} s_i \right\} \tag{10}$$

$$\text{subject to} \quad \sum_{e \in \sigma_i} x_e \le \sum_{k=1,\ldots,K} a_k y_i^k + s_i, \quad \forall i \in V, \tag{11}$$

$$\sum_{k=1,\ldots,K} y_i^k \le 1, \quad \forall i \in V, \tag{12}$$

$$\sum_{i \in V} y_i^k \le g_k, \quad \forall k = 1, \ldots, K, \tag{13}$$

$$x_e \in \{0,1\}, \quad \forall e \in E, \tag{14}$$

$$y_i^k \in \{0,1\}, \quad \forall k = 1, \ldots, K, i \in V, \tag{15}$$

$$s_i \ge 0, \quad \forall i \in V. \tag{16}$$

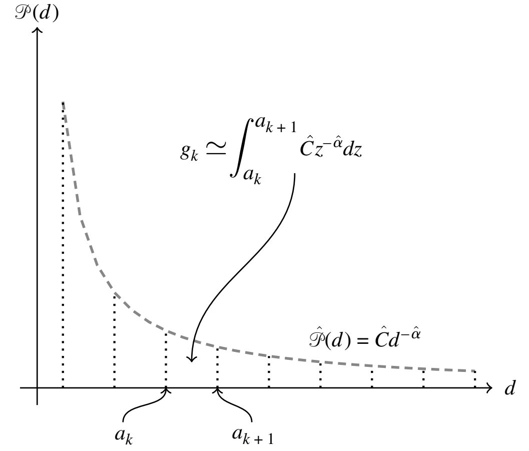


Figure 1 **Approximation of the Node Degree Function**
Note. Note that the function $\hat{\mathscr{P}}(d)$ is actually a straight line in the log-log plot.

The variables s_i for all $i \in V$ are introduced for relaxing the node degree restriction; i.e., we penalize the deviation from the expected node degree distribution in the predicted network. The parameter D controls the degree of relaxation of the node degree distribution constraints of the predicted networks. Constraints (11) restrict the node degree values of the nodes. No node can belong to more than one node degree interval, which is ensured by constraints (12). Constraints (13) ensure that the number of nodes belonging to node degree interval k should not be greater than g_k.

The number of variables of problem ($\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}}$) is $N(N-1)/2 + N \times K + N$, which may still be too large for a large-sized network to solve Problem 1 directly. Consequently, we used a simple rounding heuristic: we solve the linear relaxation of ($\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}}$)—by replacing constraints (14) and (15) with $0 \le x_e \le 1$ and $0 \le y_i^k \le 1$—and round off the (possibly) fractional solution x_e for all $e \in E$ to obtain an integer solution.

5. Experimental Study

In this section, the computational results of the proposed algorithm are reported. All algorithms were implemented using `Matlab`, and `R` was used only for the TS analysis. We used `CPLEX` to solve the optimization problem ($\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}}$).

5.1. Test Networks

We used four different networks for our computational study. The first is the Enron email data set containing the emails sent from and to employees of the Enron corporation (Cohen 2004). The second network is the stock correlations network constructed from S&P 500 companies (Kim et al. 2002). The third and fourth networks are Facebook friend networks (Viswanath et al. 2009).

5.1.1. Enron Email Network. The Enron email network is constructed from the emails sent from and to employees of the Enron corporation (Cohen 2004). Because this data set has many integrity issues, we used the clean version available online (Shetty and Adibi 2004). The data set has been previously used for link prediction studies, particularly those focusing on surveillance issues (Boginski et al. 2006, Huang and Lin 2009), and social network analysis (McCallum et al. 2005, 2007). There are 151 individuals, mostly former senior managers of Enron. The data span from May 11, 1999 to June 21, 2002, and we have created 38 networks for each month from May 1999 to June 2002. Each monthly network, which we denote G_t for any $t = 1, \ldots, 38$, has 151 nodes. A link between two nodes (individuals) is made if two individuals exchanged at least one email during the month. Our first step was to make the weighted version of adjacency matrix $\hat{M}_t$ for time period t, where the element

$\hat{m}_{ij}^t$ is the number of exchanged emails between i and j during the period t. We define the unweighted version of adjacency matrix M_t from $\hat{M}_t$: element m_{ij}^t is 1 if and only if $\hat{m}_{ij}^t > 0$. We denote $G_{t_1\sim t_2}$ as the reduced network for the multiple periods whose link set is the union of the link sets for networks $G_{t_1}, \dots, G_{t_2}$. The unweighted adjacency matrix $M_{t_1\sim t_2}$ for the reduced network $G_{t_1\sim t_2}$ is given accordingly: its (i, j) element is 1 if and only if a link exists between i and j in the reduced network $G_{t_1\sim t_2}$.

5.1.2. Stock Correlation Network. In the world of global finance, an understanding of the behavior of stock correlations between individual companies is of critical importance. Consequently, many studies have focused on the stock correlation network in the structural analysis of a network (Boginski et al. 2006, Kim et al. 2002) and the link prediction setting (Lahiri and Berger-Wolf 2007). The stock correlation network is constructed by calculating the correlation between every two companies, where each link has the same correlation value as the weight. A certain threshold value β is commonly used to filter the strongly correlated links; i.e., the link between i and j is made only if the correlation between them is greater than or equal to β.

We consider the correlations of the daily close prices of 487 companies in the S&P 500 Index. First, for a given day l, we calculate a quantity $Q_i(l)$ of company i to eliminate the time-dependent factors that might be caused by external economic environment changes, as proposed by Kim et al. (2002):

$$Q_i(l) = S_i(l) - \frac{1}{|V|}\sum_{i\in V} S_i(l), \tag{17}$$

where $S_i(l) := \log Y_i(l+1) - \log Y_i(l)$, and $Y_i(l)$ is the (close) stock price of company i on day l. Then, for a given time period t, the weight for the link between i and j is given as

$$w_{ij}^t = \frac{\langle Q_i Q_j\rangle_t - \langle Q_i\rangle_t \langle Q_j\rangle_t}{\sqrt{(\langle Q_i^2\rangle_t - \langle Q_i\rangle_t^2)(\langle Q_j^2\rangle_t - \langle Q_j\rangle_t^2)}}, \tag{18}$$

where $\langle\cdot\rangle_t^2$ is the mean value over period t. In a similar manner to the one we used for the Enron network, we build 36 monthly networks from January 2008 to December 2010 using w_{ij}^t values calculated as the link weights. We set the value of β as 0.7, i.e., a link (i, j) in the unweighted version of the monthly network G_t at time period t is added if and only if $w_{ij}^t \geq 0.7$. The goal is to predict links that are strongly correlated (greater than or equal to 0.7) at period T based on our previous knowledge of periods $T-12, \dots, T-1$.

5.1.3. Facebook Friend Network. Because of the advance of Internet technology, a social network like Facebook is becoming increasingly popular. Unlike the Enron email network and stock correlation network, the Facebook friend network is ever growing. That is, a network at period T always completely contains edges of period $T-1$. Thus, our goal of link prediction is to predict the newly associated friend-links based on the past network information. In this study, we used the Facebook friend network data provided by Viswanath et al. (2009). The data set originally was obtained by compiling a New Orleans regional Facebook friend network. There are 63,731 distinct individuals in the data set. From the original data set, we made two data sets—Facebook500 and Facebook1000—that contain the first 500 individuals for Facebook500 and 1000 individuals for Facebook1000 and links between only them. Some links in the original data set have the time of link establishment. We first constructed a base network G_0 having links that do not have the time information. The base networks for Facebook500 and Facebook1000 have 2,246 and 6,017 edges, respectively. We then created networks for every two months having the newly created links only during that period. As a result, we have 14 networks $(G_1, \dots, G_{14})$ spanning from September 2006 to December 2008. The goal is to predict the newly associated friend-links at time T from the information of networks $G_0, G_1, \dots, G_{T-1}$. Let $\hat{E} \subseteq E$ denote the set of edges created before period T. For the SO methods, we first choose all edges in $\hat{E}$ and then take n^* edges with top score values among the remaining edges. For the DD approaches, we first fixed variables x_e for all exiting edges by adding constraints $x_e = 1$, $\forall\, e \in \hat{E}$ to problem $(\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}})$. Except fixing of variables, the algorithm is the same to the cases of Enron email networks and stock correlation networks.

5.2. Baseline Methods and Performance Evaluation

For a given period T, we prepared a reduced (unweighted) network $G_{T-12\sim T-1}$ from the previous 12 monthly networks for the Enron email network and stock correlation network. For the Facebook network, we aggregated all past networks $G_0 \sim G_{T-1}$ that represent the topology of the network just before the time of prediction. We then calculated the probability (score) matrix from the reduced network with the result that each score matrix for any period T contains the aggregated information of the past networks.

In this study, we tested the following four scoring methods:

Static Scoring Method (*ST*). From the reduced graph of the past networks, we calculate the static score matrices S_{ADA}, S_{KZ}, and S_{PA} using the scoring algorithms ADA, KZ, and PA, respectively. We normalize each scoring matrix by dividing it by the maximum score of each score matrix. The static score matrix S_{ST} can then be given as the sum of all score matrices; i.e., $S^{\mathrm{ST}} := (S_{\mathrm{ADA}} + S_{\mathrm{KZ}} + S_{\mathrm{PA}})/3$.

TS Scoring Method. To obtain the score matrix S_{TS} for the TS approach, we follow the method proposed by Huang and Lin (2009): for each link, the ARIMA(p, d, q) fitting for $p = 0, 1, 2, 3$, $d = 0, 1$, and $q = 0, 1, 2, 3$ is performed from the TS data of the link weights of the previous 12 months (i.e., $\{\hat{m}_{ij}^{T-12}, \ldots, \hat{m}_{ij}^{T-1}\}$). The median $\hat{x}_{ij}$ and the standard deviation sd_{ij} of the TS data are obtained from the best ARIMA model for each link. For the Enron network, the TS data correspond to the occurrence frequency of exchanged emails between two nodes. The score for link (i, j) is given as $\Pr(\hat{x}_{ij} > 1)$, which is the probability of at least one email existing at period T. For the stock correlation network, the TS data consist of the correlation values of the previous 12 months, which have values between -1 and 1. The score is given as $\Pr(\hat{x}_{ij} > 0.7)$, which represents the probability that the correlation at period T is greater than 0.7. For the Facebook networks, we cannot adopt this TS method because each link does not constitute TS data; i.e., a link is ever existing if it was created once before.

Hierarchical Random Graph Model (*HRM*). From the aggregated past network, we calculated the connection probability p_{ij} for every link in the network by using the HRM proposed by Clauset et al. (2008). We used the computer code provided at http://www.santafe.edu/~aaronc/randomgraphs/. The original implementation does not calculate the probabilities for the existing links. We modified the code to produce the probabilities for the existing links because for the Enron email networks and stock correlation networks the links observed in the past networks can disappear in the future networks. Let S_{HRM} denote the probability matrix whose (i, j) element represents the probability of link (i, j).

Hybrid Scoring Method (*ALL*: *ST+TS+HRM*). This score matrix tries to combine the static information, the TS aspect of the targeted network, and the link probability based on HRM by summing three score matrices S^{ST}, S_{TS}, and S_{HRM}; i.e., $S_{\mathrm{ALL}} := (S^{\mathrm{ST}} + S_{\mathrm{TS}} + S_{\mathrm{HRM}})/3$. Since the Facebook networks do not have TS information, we used $S_{\mathrm{ALL}} := (S^{\mathrm{ST}} + S_{\mathrm{HRM}})/2$ for the Facebook networks.

Note that the score matrix obtained by each of these methods may contain many zero scores. To prevent the appearance of these zero scores, a small value $s_{\min}/2$ was added to each element of the score matrix, with $s_{\min}$ being the smallest nonzero score value of the score matrix.

The performance of each algorithm was measured by a receiver operation characteristics (ROC) curve (Bradley 1997). A ROC curve summarizes the predictive performance of the algorithm by relating the percentage of true positive predictions (= sensitivity, y-axis) to the percentage of false positive predictions (= 1−specificity, x-axis). Therefore, the ROC curve is a two-dimensional plot where the x and y axes range from 0 to 1. A random prediction algorithm should produce a straight line connecting the left-bottom corner $(x, y) = (0, 0)$ and upper-right corner $(1, 1)$. The most desirable prediction algorithm is the one that can produce a ROC curve closer to the upper-left corner $(0, 1)$. Once the ROC curve is obtained, we can calculate the area under the curve (AUC) value from the plot. The AUC value clearly ranges between 0 and 1, with the perfect prediction algorithm having an AUC value of 1 and the random algorithm having an AUC value of approximately 0.5.

For the SO algorithms, the ROC curve was obtained by solving the $(\mathrm{P_{SO}})$ problem repeatedly for the given score matrix, increasing the values of n^*. In contrast, for the $(\mathrm{P_{DD}^R})$ problem, the ROC curve was obtained by increasing the minimum and maximum values of node degrees of the predicted network until the true positive rate was greater than 95%. After solving the linear relaxation problem of $(\mathrm{P_{DD}^R})$ for the given degree intervals, we simply rounded off the (possibly) fractional solution to obtain an integer solution.

In addition to obtaining AUC values, we also compare the performance of the algorithms by using the performance profile graphs proposed by Dolan and Moré (2002). In comparing the performance of different algorithms on the different sets of problems, we may find that the simple average value may be biased for certain specific problem cases. The performance profile graph may eliminate these undesirable biases and provide a simple and concise representation of the relative performance of the different algorithms. Thus, for a set of algorithms A and a set of problems P, we define the *performance ratio* as

$$r_{p,a} = \frac{\max_{a\in A}\{t_{p,a}\}}{t_{p,a}}, \quad \text{for all } p \in P, \tag{19}$$

where $t_{p,a}$ is any performance measure (AUC value in this study) of algorithm $a \in A$ for problem $p \in P$. We then define $\rho_a(\tau)$ as the probability for algorithm $a \in A$ that a performance ratio is within a factor τ of the best possible ratio:

$$\rho_a(\tau) = \frac{|\{p \in P \mid r_{p,a} \le \tau\}|}{|P|}. \tag{20}$$

For any given $\tau \ge 1$, algorithms having large $\rho_a(\tau)$ are preferred. In particular, $\rho_a(1)$ represents the probability that the algorithm a will not be outperformed by the rest of the algorithms. A performance profile graph is obtained by plotting the probability $\rho_a(\tau)$ with varying τ.

There are two methods of link prediction (SO and DD) and four scoring methods (ST, TS, HRM, and ALL). We denote X_{Y} if the prediction is made by prediction method X and score matrix Y. For example, $\mathrm{DD}_{\mathrm{ST}}$ stands for the DD approach with the S^{ST} scoring matrix. Similarly, $\mathrm{SO}_{\mathrm{ALL}}$ means that the combined score matrix S_{ALL} is used in the SO method. Consequently, eight distinct algorithms are used in this study: $\mathrm{SO}_{\mathrm{ST}}$, $\mathrm{SO}_{\mathrm{TS}}$, $\mathrm{SO}_{\mathrm{HRM}}$, $\mathrm{SO}_{\mathrm{ALL}}$, $\mathrm{DD}_{\mathrm{ST}}$, $\mathrm{DD}_{\mathrm{TS}}$, $\mathrm{DD}_{\mathrm{HRM}}$, and $\mathrm{DD}_{\mathrm{ALL}}$.

We build S^{ST} by combining three well-known static scoring methods with the same weight. It is also possible to use different weights in combination, which may enable us to find an *optimal* combination of weights. However, determining an optimal combination of weights is nontrivial and may need sophisticated methods such as ensemble of classifiers (Polikar 2006), which is not a major concern of this paper. Table 1 shows AUC values for Enron email networks using three single-scoring methods ($\mathrm{SO}_{\mathrm{ADA}}$, $\mathrm{SO}_{\mathrm{KZ}}$, and $\mathrm{SO}_{\mathrm{PA}}$) and one combined method ($\mathrm{SO}_{\mathrm{ST}}$). Boldface font indicates the best results among algorithms. It is evident that finding a single optimal combination weight is not straightforward because the best scoring method is quite different for each network instance. At least the equal weights ($\mathrm{SO}_{\mathrm{ST}}$) method used in this study showed not only the best average result but also the most stable result (i.e., the smallest standard deviation).

Table 1 **Link Prediction Results (AUC Values) of Static Scoring Methods (Enron Email Networks)**

Month-year	$\lvert E_t\rvert$	$\hat{\alpha}$	$\mathrm{SO}_{\mathrm{ADA}}$	$\mathrm{SO}_{\mathrm{KZ}}$	$\mathrm{SO}_{\mathrm{PA}}$	$\mathrm{SO}_{\mathrm{ST}}$
5-2000	46	1.51	0.8450	**0.9542**	0.9457	0.9585
6-2000	66	1.51	0.7793	**0.8695**	0.8572	0.8698
7-2000	82	1.51	0.7417	**0.8892**	0.8471	0.8826
8-2000	129	1.42	0.8029	0.8823	0.8502	**0.8844**
9-2000	100	1.40	0.8851	**0.9719**	0.9116	0.9648
10-2000	141	1.38	0.8494	**0.8867**	0.8248	0.8804
11-2000	165	1.34	0.8633	**0.9614**	0.8566	0.9507
12-2000	164	1.45	0.8941	**0.9602**	0.8434	0.9455
1-2001	148	1.61	0.9021	**0.9348**	0.8410	0.9258
2-2001	172	1.58	0.9200	**0.9674**	0.8658	0.9606
3-2001	184	1.50	0.9528	**0.9743**	0.8769	0.9700
4-2001	212	1.44	0.8887	**0.9165**	0.8032	0.9039
5-2001	249	1.36	0.7868	**0.8240**	0.7288	0.8163
6-2001	197	1.34	0.8186	0.8131	0.6564	**0.8146**
7-2001	219	1.41	0.8287	**0.8386**	0.6877	0.8381
8-2001	342	1.38	0.8441	0.8509	0.6743	**0.8534**
9-2001	303	1.29	**0.8990**	0.8766	0.7103	0.8800
10-2001	490	1.33	**0.8898**	0.8547	0.6910	0.8679
11-2001	410	1.13	**0.9023**	0.7566	0.6926	0.8358
12-2001	279	1.04	**0.9208**	0.7235	0.7194	0.8594
1-2002	275	0.99	**0.8938**	0.6995	0.6822	0.8295
2-2002	246	1.09	**0.8472**	0.6899	0.7100	0.8245
3-2002	72	1.29	**0.9304**	0.3280	0.6185	0.8298
Average	204.0	1.36	0.8646	0.8445	0.7780	**0.8846**
Std-dev			0.0522	0.1388	0.0921	**0.0515**

5.3. Node Degree Distribution Estimation

As shown in the previous study, which revealed the existence of the power law node degree distribution, a power law node degree distribution appears in the log-log plot as a straight line with a negative slope (Shetty and Adibi 2004). To estimate the value of α for each monthly network at period T, we applied the least-square linear fitting using the function

$$\log P(d) = C - \alpha \log d, \tag{21}$$

where the fitting is conducted over the aggregated node degree histograms of the past networks; 12 monthly networks $G_{T-12}, \ldots, G_{T-1}$ for the Enron email networks and stock correlation networks; and all past networks $G_0, \ldots, G_{T-1}$ for the Facebook friend networks. The value of α varies slightly over time due to environmental changes in the networks. For example, in the Enron email networks, the values of α range from 0.99 (January 2002) to 1.61 (January 2001). Figure 2 shows examples of node degree distributions and the fitted lines for four tested networks. It is noteworthy that we estimate the value of α from the past networks because we do not know the value of α of the future networks.

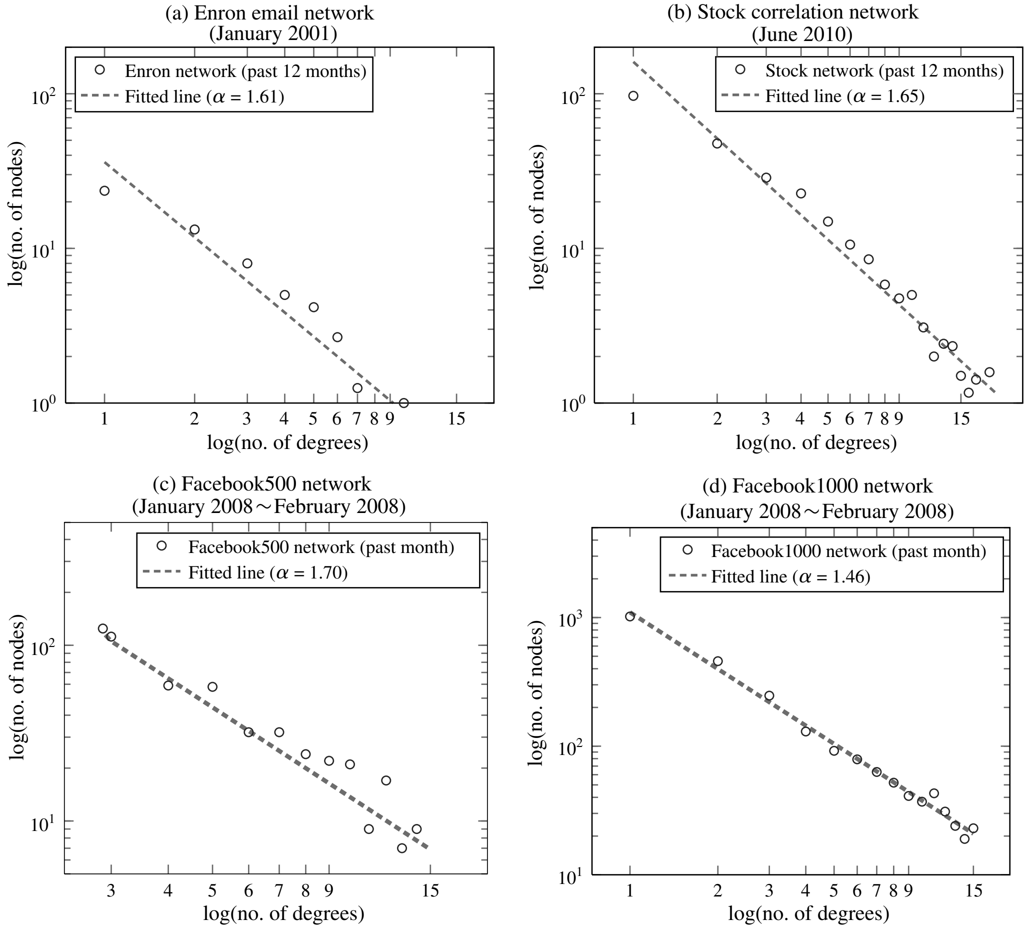
In the scale-free networks, the constant C can vary according to the maximum and minimum degree values of the network. Let l and u denote the minimum and maximum degree value, respectively. From the axiom of the probability ($\int_l^u \hat{P}(z)dz = 1$), the value of $\hat{C}$ can then be given as follows:

$$\hat{C} = \frac{1-\hat{\alpha}}{u^{1-\hat{\alpha}} - l^{1-\hat{\alpha}}}. \tag{22}$$

Note that without the scale-free property, it is not straightforward to determine the parameter of degree distribution. For example, if the degree distribution follows a normal distribution, we need to determine the mean and standard deviation of degree values, which is not trivial, because the mean and standard deviation of degree values are changing with more prediction of links. That is, the parameters of degree distribution may change with network size, which means using the distribution as an invariant characteristic is not possible.

5.4. Choice of Algorithmic Parameters

For solving the ($\mathrm{P}_{\mathrm{DD}}^{\mathrm{R}}$) problem, two algorithmic parameters should be determined. The number of node degree intervals K should be decided by taking account of trade-off between computational time and better approximation of degree distribution. For the penalty of deviations from degree distribution, using 1 as the value of D seems a reasonable choice. Because we have normalized the score matrices, any score of an edge can be at most 1. Setting $D = 1$ implies



(a) Enron email network (January 2001)

(b) Stock correlation network (June 2010)

(c) Facebook500 network (January 2008 ∼ February 2008)

(d) Facebook1000 network (January 2008 ∼ February 2008)

Figure 2 (Color online) Examples of Node Degree Distributions in the Log-Log Plot

that if any edge has really high probability (∼ 1) the degree constraint for that edge can be relaxed by one. Table 2 shows average performances of algorithm DD_{ALL} using different parameters for Enron email networks. The column "Time (sec)" represents time spent in solving one network. As expected, with increasing K we need more time to solve the problem. Using value of D greater than 1 did not produce any difference in the results. All results in the following were obtained using $K = 9$ and $D = 1$.

5.5. AUC Results for Tested Networks

Table 3 summarizes the performance of various algorithms for the networks tested. The best AUC values are shown in bold. The column $\hat{\alpha}$ represents the used value of α for the degree distributional approaches. Note that these values are not of the targeted networks but an aggregation of past 12 networks. (The online supplement has more detailed result tables; see Tables B1–B4 therein.)

Table 2 Average AUC Values of DD_{ALL} for Enron Email Networks with Different Algorithmic Parameters

	D					
K	0.1	0.5	1	5	10	Time (sec)
3	0.9014	0.9077	0.9078	0.9078	0.9078	6.0
9	0.9021	0.9080	**0.9081**	0.9081	0.9081	9.2
15	0.9021	0.9084	0.9079	0.9079	0.9079	11.9

5.5.1. Enron Email Networks. All scoring methods except HRM improved the performances when used with the proposed algorithm. The DD_{ALL} algorithm showed the best performance, and the SO_{ALL} and DD_{ST} algorithms showed comparable performances. Note that our node degree restriction algorithm actually tends to suppress the prediction of high-scored links compared with the simple ordering algorithms. For example, when the degree distributional algorithm and the simple ordering algorithm predict the same number of links for the future network, the simple ordering algorithm is better than (or equal to) the DD algorithm in terms of the total score sum of links predicted. However, the results also clearly show that simple maximization of the sum of scores does not necessarily yield a better prediction performance. SO_{ST} (DD_{ST}) performed considerably better than SO_{TS} (DD_{TS}). This behavior is

Table 3 Link Prediction Results (AUC Values) for the Networks Tested

			Static		Time-series		HRM		All	
	$\lvert E_t \rvert$	$\hat{\alpha}$	SO_{ST}	DD_{ST}	SO_{TS}	DD_{TS}	SO_{HRM}	DD_{HRM}	SO_{ALL}	DD_{ALL}
Enron	204.0	1.36	0.8846	0.8998	0.8378	0.8423	0.8729	0.8686	0.9065	**0.9081**
Stock	1,121.8	1.38	0.8395	0.8530	0.8309	0.8286	0.8743	0.8641	**0.8788**	0.8751
Facebook500	40.0	1.57	0.8709	0.9031			0.8728	0.8716	0.8975	**0.9104**
Facebook1000	114.3	1.32	0.8543	**0.8927**			0.7697	0.7672	0.8522	0.8855

understandable because the TS analysis cannot produce a meaningful prediction when there are no previous links in the past 12 monthly networks.

5.5.2. Stock Correlation Network. The DD_{ALL} and SO_{ALL} algorithms showed the best performance in terms of average AUC value. It is interesting to note that the DD algorithms appear to perform better for networks with small target links (see Table B.2 in the online supplement). For example, the best p-value is obtained for networks having fewer than 500 edges by the static scoring method (DD_{ST} versus SO_{ST}). For a given monthly network, let $\mathcal{P}$ and $\hat{\mathcal{P}}$ denote *real* and *estimated* degree distributions, respectively. Because $\hat{\mathcal{P}}$ was obtained from past 12 monthly networks not including the current network, it is not necessarily the same with $\mathcal{P}$, which is obtained from the current network only. Note that our method relies on the assumption of persistence of degree distribution property, which means the actual performance of our algorithm may depend highly on similarity of degree distributions such as Kullback-Leibler (KL) divergence. Let $\Delta_{\text{ST}} := \text{DD}_{\text{ST}} - \text{SO}_{\text{ST}}$ and $\Delta_{\text{ALL}} := \text{DD}_{\text{ALL}} - \text{SO}_{\text{ALL}}$, e.g., performance gain from our method. Figure 3 shows correlation between Δ_{ST} (or Δ_{ALL}) and $\text{KL}(\mathcal{P}, \hat{\mathcal{P}})$, where clear negative correlation is shown (e.g., -0.65 and -0.75). This result strongly indicates that accurate estimating of degree distribution is important for performance of our method, and our method performs better when the distribution is similar enough.

5.5.3. Facebook Networks. DD_{ALL} showed the best performance for the Facebook500 network, and DD_{ST} performed best for the Facebook1000 network. In both cases, the performance of SO_{ST} and SO_{ALL} was greatly improved by the DD approach. Comparing the two results, it is notable that the HRM method did not perform well, especially for the Facebook1000, whereas other methods could produce more consistent results. The HRM method relied on hierarchical decompositions (dendrograms) of the given network. And the possible number of dendrograms of the network is growing exponentially with the network size. Since trying all possible dendrograms is practically impossible, a fixed number of dendrograms having good likelihood are sampled. Consequently, with an increasing network size, the HRM method may suffer from the lack of sampled dendrograms, which results in poor performance.

5.6. Analysis of Results

Figure 4 contains plots of the performance profile graphs of the tested algorithms for the four tested networks. In these graphs, the y-axis is the probability that any algorithm would perform as well as the

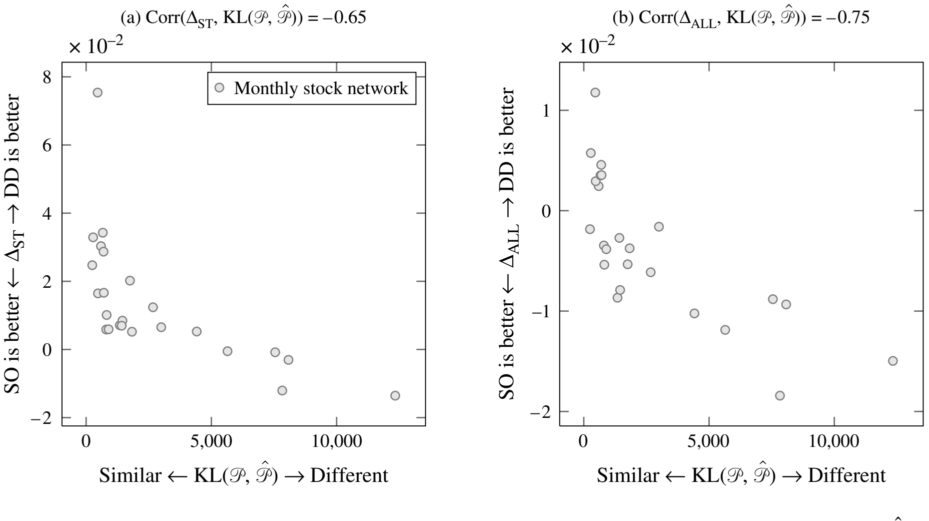


Figure 3 (Color online) Correlation Between Performance of Our Algorithm and Similarity of Degree Distribution ($KL(\mathcal{P}, \hat{\mathcal{P}})$)

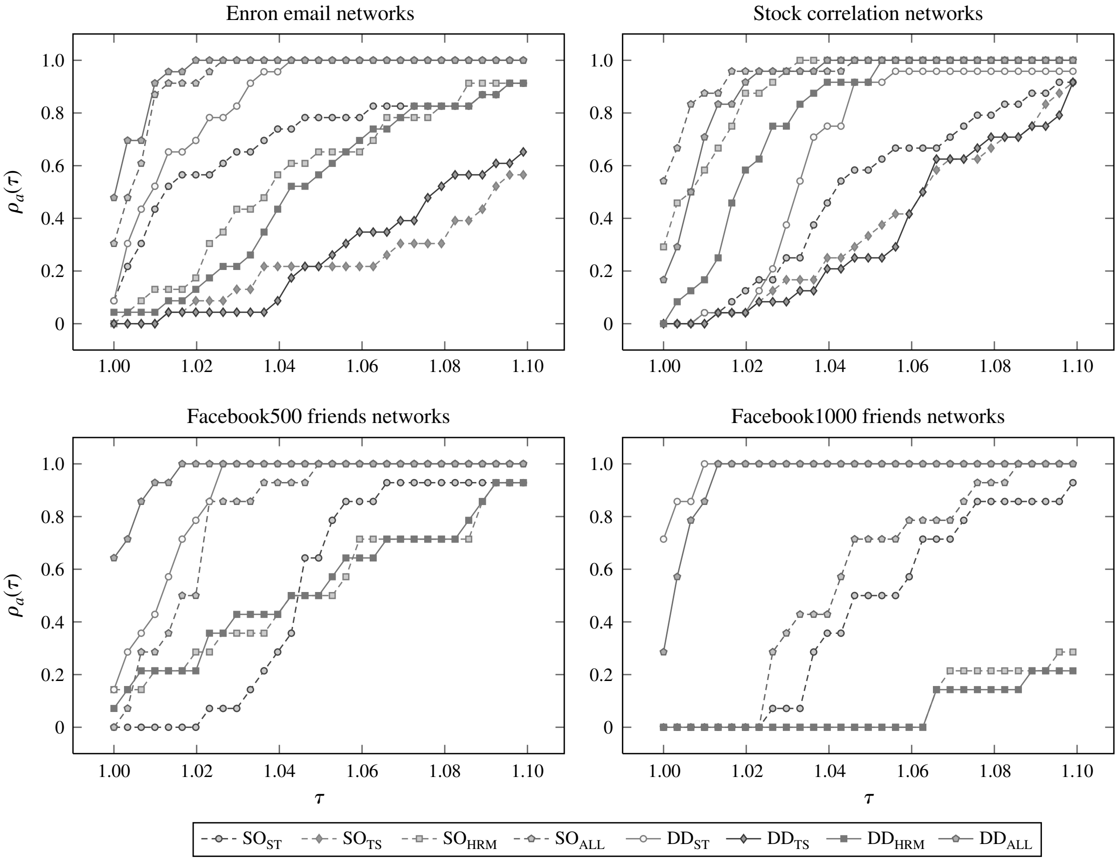


Figure 4 (Color online) Performance Profile Graph for the Tested Networks

best possible algorithm with the ratio of $1/\tau$. In other words, a point $(\hat{\tau}, \hat{\rho})$ indicates that the performance of the algorithm would be within $1/\hat{\tau}$ of the best possible algorithm with probability $\hat{\rho}$. Therefore, a good algorithm is expected to have high ρ values for every range of τ. The performance profile graphs in Figure 4 show that DD_{ALL} is the best algorithm except for the stock correlation network case.

The additional performance boosts by our algorithm were most significant with the static scoring method (DD_{ST} versus SO_{ST}). For the TS scoring or HRM methods, however, the performance improvement of our algorithm was not significant because these methods may produce many zero scores. For example, a link of any two individuals who have never exchanged an email has zero score by the TS scoring method; however, these two individuals may be connected in an indirect way so that the graph theoretic measure, such as (KZ), gives a nonzero score. Table 4 shows the ratios of nonzero values of score matrices for the different scoring methods. The very small percentages of nonzero scores explain why the DD approach did not perform well for these scoring methods. Roughly speaking, the DD approach finds a solution having the most similar degree distributional characteristic compared to the past networks among many network solutions having similar sum of scoring values. For the DD approach, the score matrix should have sufficient nonzeros so that many alternatives of solutions can be considered. Figure 5 illustrates ROC curves for the selected networks. The plots clearly show that the DD approach can derive a benefit from the many nonzero scores of the static scoring method.

Another interesting thing to note is that our approach does not seem to improve the performance of the HRM method, despite, as shown in Table 4, this method providing considerably more nonzeros than the TS method. In their original paper proposing the HRM method, Clauset et al. found that the HRM method successfully reconstructs the statistical properties of the network closely, including degree distribution, despite the fact that their algorithm does not explicitly exploit these properties (Clauset et al. 2008).

Table 4 Ratios of Nonzero Edge Scores for Different Methods

Network	ST (%)	TS (%)	HRM (%)
Enron	66.44	5.29	49.79
Stock	98.81	7.48	46.05
Facebook500	97.61	n/a	14.16
Facebook1000	95.56	n/a	10.47

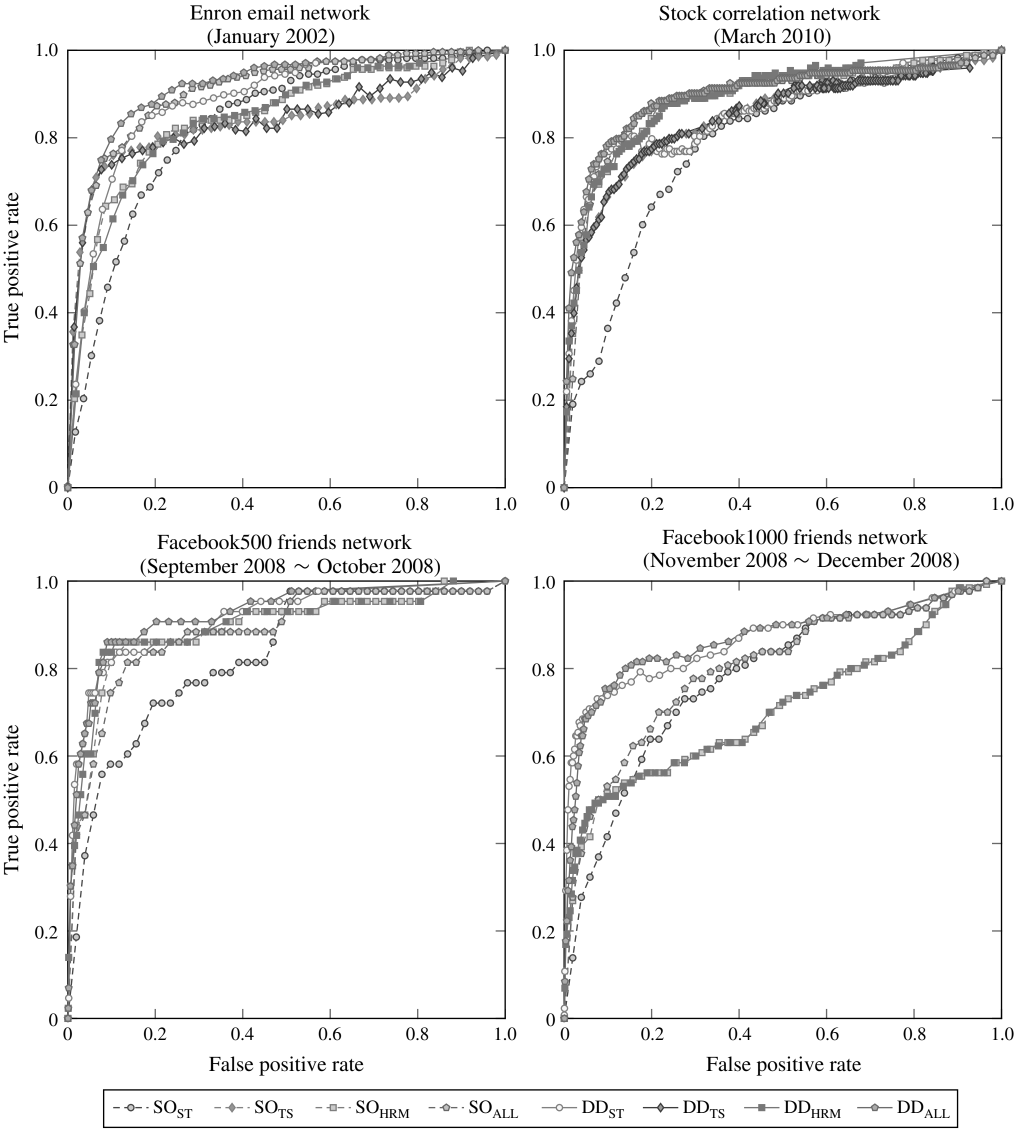


Figure 5 (Color online) ROC Curves for the Selected Networks

This explains why the HRM method does not benefit from our approach.

5.6.1. Quality of Node Degree Distribution Estimation. It is obvious that the performance of our approach depends highly on the quality of estimated node degree distribution. In the previous sections, the computational results were obtained using a simple moving average to estimate the node degree distribution. Since precise estimation of node degree distribution is very important in our approach, more sophisticated estimation methods can be employed. If we push this idea further, the best possible case would be when we use the *true* node degree distribution at time T. We conducted additional experiments to assess the prediction performance when using the true degree distribution as well as to investigate other degree distribution estimation methods. Let DD$_{\text{EXACT}}$ denote our approach with the exact (true) node degree distribution at time period T. Let N_T be the number of links to predict in time T. For the exact degree distribution, instead of fitting the distribution to any predefined distribution function, we used the true histogram of node degrees at time T in problem ($\text{P}^{\text{R}}_{\text{DD}}$). For a more sophisticated degree distribution estimation method, we conducted ARIMA(p, d, q) fitting for $p = 0, 1, 2, 3$, $d = 0, 1$, and $q = 0, 1, 2, 3$ for estimating the parameters of power law degree distribution (DD$_{\text{ARIMA}}$). DD denotes our approach with the simple moving average method used in the previous sections. With each method, we made a prediction of N^T links and calculated true positive ratios ($TP :=$ no. of correct predictions$/N^T$). For the easy comparison, we normalized the results by dividing with results of SO

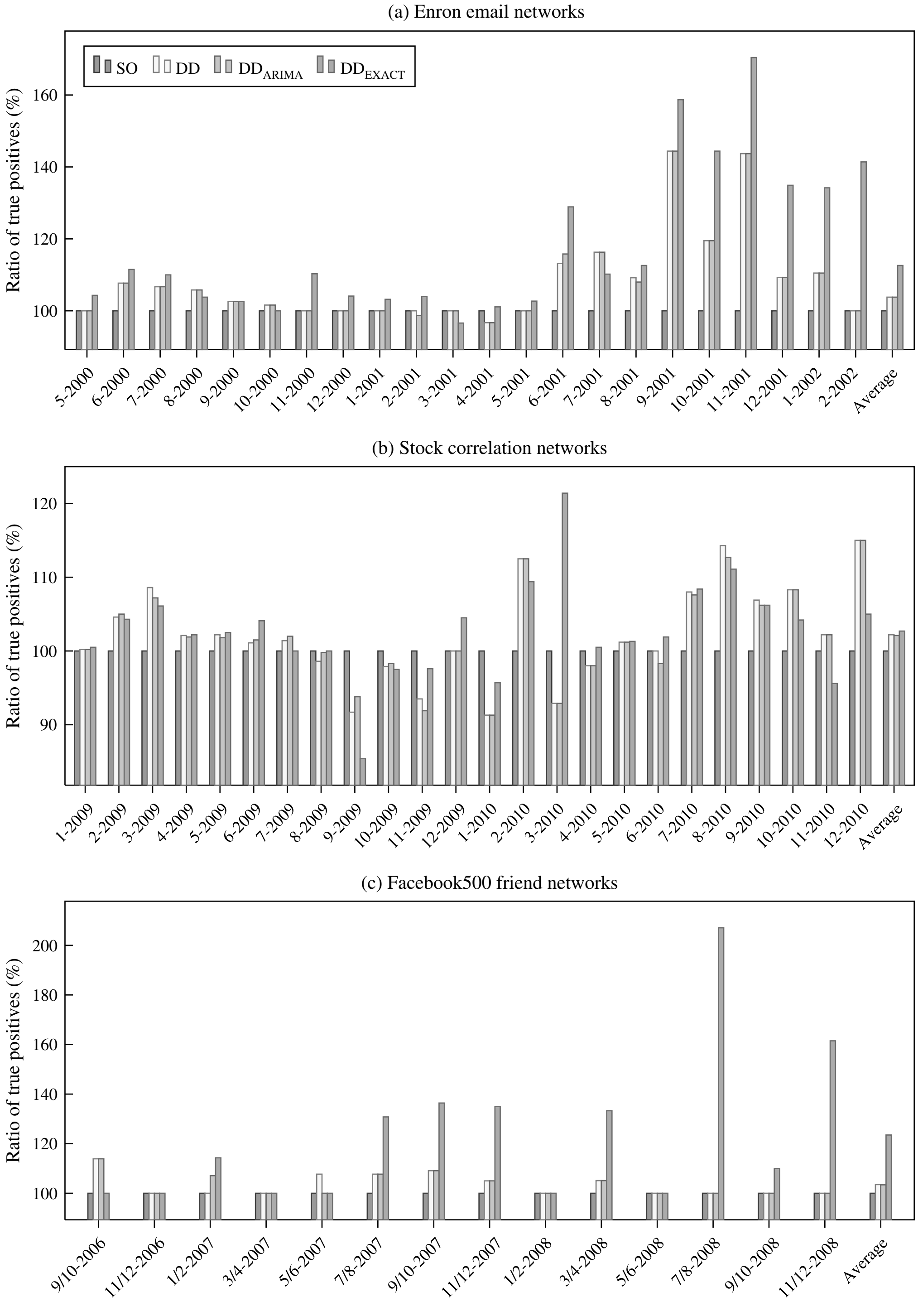


Figure 6 (Color online) Comparison of Different Node Degree Estimations

method, i.e., $TP(\mathrm{DD})/TP(\mathrm{SO})$, $TP(\mathrm{DD}_{\mathrm{ARIMA}})/TP(\mathrm{SO})$, $TP(\mathrm{DD}_{\mathrm{EXACT}})/TP(\mathrm{SO})$. Thus higher values are desirable because it means the method performs better than the SO method. We used S^{ST} for all experiments. Figure 6 shows the prediction performances of our approach with different node degree estimation methods relative to that of the SO approach. The figure clearly shows that (i) $\mathrm{DD}_{\mathrm{EXACT}}$ outperforms other methods particularly for Enron email networks and Facebook friend networks, which implies that complying with the degree distribution can improve the link prediction performance when the degree distribution can be estimated accurately; and (ii) $\mathrm{DD}_{\mathrm{ARIMA}}$ and DD are quite comparable in terms of performance, which implies that the node degree distribution estimated by the simple moving average was quite comparable to that obtained by the ARIMA fitting method.

5.6.2. Using General Degree Distribution. In general, our approach can be applied to any networks as long as the estimated node degree distribution is available, which implies that any type of node degree distribution can be used. For example, we can estimate two parameters (mean and standard deviation) from the historical data if we believe that the node degree distribution follows the normal distribution. It is even possible to use a completely arbitrary degree distribution, as we did for $\mathrm{DD}_{\mathrm{EXACT}}$ in §5.6.1. Figure 7 compares the prediction performance of different distribution functions fitted with the *true* degree distribution tested on the stock correlation networks. Let $\mathrm{DD}^{X}_{\mathrm{EXACT}}$ denote the degree distributional approach with probability distribution function X fitted to the true degree distribution at period T. The distributions used for comparison include normal, Poisson, logistic, exponential, and power law distributions. We compared the relative performance of each distribution to that of $\mathrm{DD}_{\mathrm{EXACT}}$. The results clearly show that $\mathrm{DD}^{\mathrm{Power}}_{\mathrm{EXACT}}$ performs better than other distributions. One interesting thing to note is that for some networks, $\mathrm{DD}^{\mathrm{Power}}_{\mathrm{EXACT}}$ outperforms $\mathrm{DD}_{\mathrm{EXACT}}$. These somewhat surprising results can be explained when we take into account the true positive results of $\mathrm{DD}_{\mathrm{EXACT}}$ (the upper graph of Figure 7), where lower values mean the prediction was not so successful (e.g., 9-2009, 12-2009, and 1-2010, etc.). The low true positive values also imply that the score matrix used for prediction failed to be relevant to the true links in period T. In other words, many links in period T are actually unexpected connections that could not be foreseen by the scores obtained from the past networks. Therefore, constraining with the true degree distribution by $\mathrm{DD}_{\mathrm{EXACT}}$ may impair the prediction performance because the exact degree constraint can make the prediction completely wrong with poor quality scores. In contrast, $\mathrm{DD}^{\mathrm{Power}}_{\mathrm{EXACT}}$ tends to reduce the risk of wrong prediction because the fitting effectively makes the distribution smoother and less extreme.

It should be noted that the power law assumption can give us one unique advantage over other degree distribution assumptions. Taking the power law distribution assumption implies that we also maintain the scale-free assumption, which enables us to use the same estimated value for the exponent parameter α, regardless of the number of links to predict. With other distribution assumptions, such as the normal distribution, we need to estimate distribution parameters for every possible number of links to predict. For example, consider the case that we estimated the degree distribution with two parameters μ and σ, based on the normal distribution assumption, from a past network containing Z links. The problem is that we do not know how many links will exist in the future network, and the degree distribution constraints would only make sense when the future network has the same number of links; i.e., μ and σ are only valid in a network with Z links. To predict a network with $2Z$ links, the degree distribution should be re-estimated from the past networks, which may be tricky when a network with $2Z$ links was not observed before. To overcome this, we need to

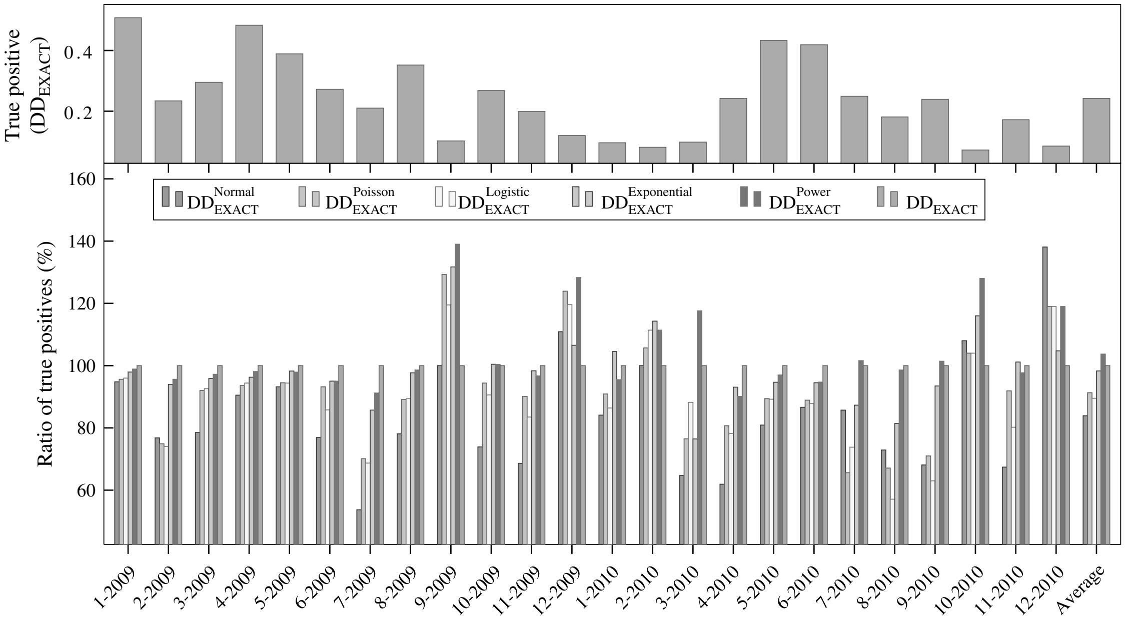


Figure 7 (Color online) Comparison of Different Distribution Functions Fitted to True Degree Distribution for Stock Correlation Networks

make other assumptions on evolution of degree distribution with every possible number of links, which requires further justifications. Under the scale-free assumption, however, it is easy to calculate the other parameter with the given exponent α and number of links 2Z because the mechanism governing evolution of the scale-free networks guarantees the same or a not significantly different α with the number of links (Barabási and Albert 1999).

Development of DD link prediction approach can be applied to general degree distributions with any number of links to predict is certainly an interesting research direction that deserves serious investigations. Another potentially interesting topic is identifying additional global structural properties of networks that may be incorporated in the link prediction algorithms. Recent studies have revealed that there are other types of global structures in real-life networks. Palla et al. (2005, 2007), for example, discovered that the size of communities in many real-life networks often follows the power law distribution. Applying this characteristic to the link prediction or the community detection problem may be valuable future research topics.

6. Conclusion and Discussion

In this paper, we propose a novel approach to the link prediction problem that exploits a network-wide characteristic to improve prediction accuracy. Traditional link prediction algorithms, such as topological inference and TS analysis, are based on some value of the likelihood measure of each single link. Although these algorithms are relatively simple and often perform well, they fall short when the collective characteristics among many links are considered. More recently, a large number of studies on real-world networks have revealed the existence of the power law of the node degree distribution. Its existence indicates that the network is scale-free, i.e., that the power law will hold regardless of the size of the network. We have developed a mathematical programming formulation that constrains the resulting link prediction solution to follow the estimated node degree distribution. We also present a new method to estimate the node degree distribution of the future network based on observations in past networks.

We tested our algorithm using three real-world networks. Although these three data sets are taken from fundamentally different types of social networks (emails between company employees, correlations between stocks, and association of friends, respectively), we were able to clearly demonstrate that each node degree distribution of each network follows a power law. The computational results show that our approach yielded a better performance than the traditional algorithm with the same scoring method. These results are rather surprising since the added performance boost can be obtained without introducing a new elaborated scoring method. We believe one of the most appealing features of our method is that it can be used in conjunction with any scoring method.

As shown in the case of the stock correlation network, an immediate extension of the proposed algorithm is to use a more elaborated approximation of the node degree distribution function. The existence of the power law is actually not an essential requirement for the application of our method. However, one major issue in using a general degree distribution is that we cannot guarantee that the degree distribution will remain the same with increasing of edges of the network. The crucial property of a power law degree distribution is its scale-freeness that is absent in general degree distribution functions. How to handle the change of general degree distributions for a growing network is a future research topic worth investigating.

The proposed method does not incorporate the temporal factors directly. One way to incorporate the temporal aspects is to treat the degree numbers of monthly networks for each node as TS data. We may predict the degree number of a node for a future network using ARIMA fitting. The obtained degree prediction does not necessary follow the global degree distribution because the ARIMA fitting should be done for each node independently. Let $\tilde{b}_i$ denote the predicted degree number for node i. We then consider the following problem:

$$
\begin{aligned}
(\mathrm{P}^{\mathrm{R}}_{\mathrm{DDT}})\quad \text{maximize}\quad & \Bigg\{\sum_{e\in E} s_e x_e - D\sum_{i\in V} s_i \\
& \quad - Q\sum_{i\in V}\bigg|\tilde{b}_i - \sum_{k=1,\dots,K} a_k y_i^k\bigg|\Bigg\} \\
\text{subject to}\quad & \sum_{e\in\sigma_i} x_e \le \sum_{k=1,\dots,K} a_k y_i^k + s_i, \quad \forall\, i\in V, \\
& \sum_{k=1,\dots,K} y_i^k \le 1, \quad \forall\, i\in V, \\
& \sum_{i\in V} y_i^k \le g_k, \quad \forall\, k=1,\dots,K, \\
& x_e \in \{0,1\}, \quad \forall\, e\in E, \\
& y_i^k \in \{0,1\}, \quad \forall\, k=1,\dots,K,\ i\in V, \\
& s_i \ge 0, \quad \forall\, i\in V,
\end{aligned}
$$

where $|\cdot|$ means an absolute value that can easily be linearized by introducing some auxiliary variables. The motivation of this problem is to use the predicted degree value as some guidance for the future degree value while respecting the global degree distribution.

It is noteworthy that the size of the proposed optimization problem can become very large when the network is large. In this case, any decomposition algorithm, such as the column-generation algorithm, may be considered. In the column-generation algorithm, for example, the master problem is modified to choose the best node degree vector having each node's degree value. The column-generation subproblem is to find the most beneficial node degree patterns (columns) with the dual optimal solution of the master problem. Another possible method for analyzing large networks is to use some heuristic algorithm in constructing a link prediction solution. In the genetic algorithm, the goodness of solution can be measured by introducing a fitness function. We can also define a fitness function measuring the node degree distribution of the solutions. The genetic algorithm then iteratively finds the solutions that are well fitted to the node degree distribution function.

Usually estimating degree distribution of future networks involves applying maximum-likelihood methods on the past networks. Unfortunately, it is often difficult to obtain accurate estimation by such methods when the network is evolving. Instead of fitting directly to any distribution functions, it is possible to use hidden *moments* of networks, which eventually determine how the networks evolve over time (Bickel et al. 2011). Incorporating the moments for degree distribution into the link prediction approach instead of the degree distribution might enable us to avoid difficulties in estimating the degree distribution precisely for the evolving networks.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ijoc.2014.0624.

Acknowledgments

The authors thank several anonymous referees for their numerous and very helpful comments and suggestions. The authors are grateful to Dr. Hongsuk Hu for fruitful discussions.

References

Adamic LA, Adar E (2003) Friends and neighbors on the web. *Soc. Networks* 25(3):211–230.

Al Hasan M, Chaoji V, Salem S, Zaki M (2006) Link prediction using supervised learning. *SDM06: Workshop on Link Analysis, Counter-Terrorism and Security, Bethesda, MD*.

Anstee RP (1987) A polynomial algorithm for *b*-matchings: An alternative approach. *Inform. Processing Lett.* 24(3):153–157.

Bader JS, Chaudhuri A, Rothberg JM, Chant J (2003) Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology* 22(1):78–85.

Barabási AL, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512..

Barabási AL, Jeong H, Neda Z, Ravasz E, Schubert A, Vicsek T (2002) Evolution of the social network of scientific collaborations. *Physica A—Mechanics Its Appl.* 311(3–4):590–614.

Bickel PJ, Chen A, Levina E (2011) The method of moments and degree distributions for network models. *Ann. Statist.* 39(5): 2280–2301.

Boginski V, Butenko S, Pardalos PM (2006) Mining market data: A network approach. *Comput. Oper. Res.* 33(11):3171–3184.

Box GEP, Jenkins GM, Reinsel GC (1970) *Time Series Analysis* (Holden-Day, San Francisco).

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7):1145–1159.

Chaudhuri K, Chung F, Tsiatas A (2012) Spectral clustering of graphs with general degrees in the extended planted partition model. *J. Machine Learning Res.* 2012:35.1–35.23.

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

Cohen W (2004) Enron email data set. Accessed February 2, 2015, http://www.cs.cmu.edu/~enron/.

Cook W, Pulleyblank WR (1987) Linear systems for constrained matching problems. *Math. Oper. Res.* 12(1):97–120.

da Silva Soares PR, Bastos Cavalcante Prudencio R (2012) Time series based link prediction. *Neural Networks (IJCNN), The 2012 Internat. Joint Conf., Brisbane, Australia*, 1–7.

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. *Math. Programming* 91(2):201–213.

Erdős P, Rényi A (1959) On random graphs, I. *Publ. Math. Debrecen* 6:290–297.

Goldberg DS, Roth FP (2003) Assessing experimentally derived interactions in a small world. *Proc. National Acad. Sci. USA* 100(8):4372–4376.

Herlau T, Morup M, Schmidt MN, Hansen LK (2012) Detecting hierarchical structure in networks. *2012 3rd Internat. Workshop Cognitive Inform. Processing (CIP), Baiona, Spain*, 1–6.

Hoff PD (2009) Multiplicative latent factor models for description and prediction of social networks. *Comput. Math. Organ. Theory* 15(4):261–272.

Hoff PD, Raftery AE, Handcock MS (2002) Latent space approaches to social network analysis. *J. Amer. Statist. Assoc.* 97(460): 1090–1098.

Huang Z, Lin DKJ (2009) The time-series link prediction problem with applications in communication surveillance. *INFORMS J. Comput.* 21(2):286–303.

Huang Z, Li X, Chen H (2005) Link prediction approach to collaborative filtering. *Proc. 5th ACM/IEEE-CS Joint Conf. Digital Libraries* (ACM, New York), 141–142.

Jeong H, Tombor B, Albert R, Oltvai ZN, Barabási AL (2000) The large-scale organization of metabolic networks. *Nature* 407(6804):651–654.

Jiang T, Tuzhilin A (2009) Improving personalization solutions through optimal segmentation of customer bases. *IEEE Trans. Knowledge Data Engrg.* 21(3):305–320.

Juszczyszyn K, Gonczarek A, Tomczak J, Musial K, Budka M (2012) A probabilistic approach to structural change prediction in evolving social networks. *2012 IEEE/ACM Internat. Conf., Adv. Soc. Network Anal. Mining (ASONAM)* (Bournemouth University, Poole, Dorset, UK), 996–1001.

Karrer B, Newman MEJ (2011) Stochastic blockmodels and community structure in networks. *Physical Rev. E* 83(1):016107-1–016107-10.

Katz L (1953) A new status index derived from sociometric analysis. *Psychometrika* 18(1):39–43.

Kim HJ, Kim IM, Lee Y, Kahng B (2002) Scale-free network in stock markets. *J. Korean Physical Soc.* 40(6):1105–1108.

Kolaczyk ED (2009) *Statistical Analysis of Network Data: Methods and Models* (Springer Verlag, New York).

Lahiri M, Berger-Wolf TY (2007) Structure prediction in temporal networks using frequent subgraphs. *IEEE Sympos. Comput. Intelligence Data Mining, CIDM, Honolulu, HI*, 35–42.

Lancichinetti A, Fortunato S, Kertész J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New J. Physics* 11(3):1–19.

Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *J. Amer. Soc. Inform. Sci. Tech.* 58(7):1019–1031.

Lu L, Zhou T (2011) Link prediction in complex networks: A survey. 390(6):1150–1170.

Mamitsuka H (2012) Mining from protein-protein interactions. *Data Mining Knowledge Discovery* 2(5):400–410.

Manning CD, Raghavan P, Schütze H (2008) *Introduction to Information Retrieval*, Vol. 1 (Cambridge University Press, Cambridge, UK).

McCallum A, Corrada-Emmanuel A, Wang X (2005) The author-recipient-topic model for topic and role discovery in social networks, with application to Enron and academic email. *Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA*, 33–44.

McCallum A, Wang X, Corrada-Emmanuel A (2007) Topic and role discovery in social networks with experiments on Enron and academic email. *J. Artificial Intelligence Res.* 30(1):249–272.

Medina A, Matta I, Byers J (2000) On the origin of power laws in Internet topologies. *ACM SIGCOMM Comput. Comm. Rev.* 30(2):18–28.

Newman MEJ (2001a) Clustering and preferential attachment in growing networks. *Physical Rev. E* 64(2):025102.

Newman MEJ (2001b) The structure of scientific collaboration networks. *Proc. National Acad. Sci. USA* 98(2):404–409.

Newman MEJ (2003) The structure and function of complex networks. *SIAM Rev.* 45(2):167–256.

Newman MEJ (2012) Communities, modules and large-scale structure in networks. *Nature Physics* 8(1):25–31.

Palla G, Barabási AL, Vicsek T (2007) Quantifying social group evolution. *Nature* 446(7136):664–667.

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435(7043):814–818.

Park Y, Moore C, Bader JS (2010) Dynamic networks from hierarchical Bayesian graph clustering. *PLoS One* 5(1):e8118.

Polikar R (2006) Ensemble based systems in decision making. *Circuits Systems Magazine, IEEE* 6(3):21–45.

Salton G (1989) *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer* (Addison-Wesley, Reading, MA).

Shen H, Cheng X, Guo J (2011) Exploring the structural regularities in networks. *Physical Rev. E* 84(5):056111.

Shetty J, Adibi J (2004) The Enron email data set database schema and brief statistical report. Technical report, Information Sciences Institute, University of Southern California.

Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in Facebook. *Proc. 2nd ACM SIGCOMM Workshop Soc. Networks (WOSN'09), Barcelona, Spain*.

Zhou S, Mondragón RJ (2004) Accurately modeling the Internet topology. *Physical Rev. E* 70(6):066108.