A Novel Link Prediction Approach for Scale-free Networks

Chungmok Lee
IBM Research–Ireland
Damastown Industrial Park,
Mulhuddart
Dublin 15, Ireland
chungmok@gmail.com

Minh Pham
RUTCOR
Rutgers University
Piscataway, New Jersey, USA
ptuanminh@gmail.com

Norman Kim
RUTCOR
Rutgers University
Piscataway, New Jersey, USA
norman.kim@gmail.com

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

Dennis K. J. Lin
Department of Supply Chain
and Information Systems
Pennsylvania State University
Pennsylvania, USA
dennislin@psu.edu

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

ABSTRACT

The link prediction problem is to predict the existence of a link between every node pair in the network based on the past observed networks arising in many practical applications such as recommender systems, information retrieval, and the marketing analysis of social networks. Here, we propose a new mathematical programming approach for predicting a future network utilizing the node degree distribution identified from historical observation of the past networks. We develop an integer programming problem for the link prediction problem, where the objective is to maximize the sum of link scores (probabilities) while respecting the node degree distribution of the networks. The performance of the proposed framework is tested on the real-life Facebook networks. The computational results show that the proposed approach can considerably improve the performance of previously published link prediction methods.

Categories and Subject Descriptors

G.1 [**Numerical Analysis**]: Optimization—*Integer programming*; F.2 [**Analysis of Algorithms and Problem Complexity**]: General; I.2 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*

Keywords

analysis of algorithms; data mining; link prediction; optimization;

WWW'14 Companion, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2580049.

1. INTRODUCTION

The link prediction problem is to predict the existence of a link between every node pair in the network based on the past observed networks [17]. For example, the aim of information retrieval is to classify unidentified documents by predicting the relationships (links) between words and document classes, where each node denotes a word or a document class [23, 18]. The analysis of biological interactions is another example of scientific field in which the link prediction problem is clearly relevant primarily due to the high experimental costs for large biological networks. In [4], the authors 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. The recommender system is another important application of the link prediction problem. In [11], a number of graph theoretic measures between the users and the items were adapted 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 a preference between them.

The link prediction problem can also be applied in evolving networks also. For instance, how the structure of Internet topology is evolving over time has been an important question in computer science and social science [19, 26]. 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 of practical interest. Predicting the prospective links in the co-authorship network was investigated in [2], where the link prediction was treated as a supervised learning problem.

2. PREVIOUS ALGORITHMS

The simplest (and arguably most effective) algorithms for solving the link prediction problem are the so-called *scoring methods*. The scoring function assigns a certain score to the link, while 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. With the link scores calculated, the prediction can be made by sorting the link scores in a decreasing order and choosing a predefined number of links with top scores. A comparison of the prediction performances of many scoring methods on the co-authorship network can be found in [16].

Each algorithm for the link prediction problem which has been proposed up to now can essentially be considered as an estimation method of each link and, consequently as focusing solely on the probability of each link. 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 are *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 over-fitting issues. Like many data mining frameworks, the over-fitting issues can often be remedied by introducing some *regularization* 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. Extensive research on social networks has recently revealed that an existence of the so-called *power-law* of the node degree distribution [5, 7, 12, 15]. We develop a mathematical programming approach exploiting the node degree distribution so that the prediction phase will not be too greedy.

3. DEGREE DISTRIBUTION

The degree of a node represents the number of incident links to the node in the network. Let $\mathcal{P}(d)$ denote the probability of any node with node degree d in the network. Since the seminal work by [5], it has turned out that nearly every real-life network has a specific form of node degree distribution; the power-law degree distribution [22]. 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, where the power-law distribution appears like a straight-line with a negative slope. This implies that the degree distribution function has a form

$$\mathcal{P}(d) \propto d^{-\alpha}, \tag{1}$$

where α is a constant that varies with the network type. Any network that shows the power-law degree distribution is often referred to as the scale-free network which implicates that (i) the power-law degree distribution holds regardless of the size of the network and (ii) the power-law degree distribution property is maintained 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 power-law degree distribution.

The degree distribution may be seen as some sort of global characteristics of the network, because we naturally expect that the future graph to be predicted also follows the degree distribution observed in the past. To achieve this, a link prediction method should explicitly address a specific node degree distribution of networks in making predictions. However, there are very limited studies explicitly addressing degree distribution in the link prediction settings, though some rare exceptions are found in community detection literature. In [13], a degree-corrected stochastic blockmodel was proposed to incorporate *degree heterogeneity* of communities. in the degree-corrected stochastic blockmodel, a Kullback-Leibler divergence between p_K and p_{degree} is to minimized where p_K is the probability distribution of given blockmodel and p_{degree} is the probability distribution produced by the preferential attachment model that consequently results in a power-law degree distribution. The results showed that incorporation of degree distribution property in the stochastic blockmodel performs much better in detecting real-life community structure.

4. ALGORITHM

Let $G_t(V, E_t)$ denote the undirected graph of the network at time t, where $V := \{1, \ldots, N\}$ 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 $E_1, \ldots, E_{T-1}$. 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) that is computed using various link scoring methods. Then, in all conventional link prediction algorithms, the sets of predicted links are obtained by applying a threshold value s^*, which is equivalent to taking the top n^* scored links after ordering the links. Hereafter, 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 $s(i, j)$, which We call as a *score matrix*. Then the SO algorithms ($\mathrm{P_{SO}}$) is like solving the following problem:

$$\max_{x \in \{0,1\}^{|E|}} \left\{ \sum_{e \in E} s_e x_e \mid \sum_{e \in E} x_e \leq n^* \right\}. \tag{2}$$

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, the above problem can be solved easily by sorting all elements of the score matrix and choosing top n^* links.

We now assume that the estimated probability distribution of node degrees $\hat{\mathcal{P}}(d)$ for the network G_t obtained from the past networks $G_1, \ldots, G_{T-1}$. For some nonnegative integer vector $\hat{b}$ following the degree distribution $\hat{\mathcal{P}}$, 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 ($\mathrm{P_{DD}}$) with preserving the node degree distribution can then be stated as follows:

$$\max_{b \in B} \max_{x \in \{0,1\}^{|E|}} \left\{ \sum_{e \in E} s_e x_e \mid \sum_{e \in \sigma_i} x_e \leq b_i, \forall i = 1, \ldots, N \right\}, \tag{3}$$

where σ_i is the set of the links in E adjacent to node i. We call this problem the degree distributional approach (DD). The objective of the above problem is to find a network that maximizes the sum of link scores while respecting the node degree distribution observed in the pase networks. Note that we only restrict the *distribution* of the node degrees, not the node degree of any given node i. Also note that the inner maximization problem of ($\mathrm{P_{DD}}$) can be solved polynomial time algorithms for a maximum weight b-matching problem [10, 3]. The problem ($\mathrm{P_{DD}}$) is, unfortunately, NP-hard as shown in the followings.

4.1 Computational Complexity of ($\mathrm{P_{DD}}$)

Let

$$F(s,b) := \max_{x\in\{0,1\}^{|E|}} \left\{ \sum_{e\in E} s_e x_e \mid \sum_{e\in\sigma_i} x_e \le b_i, \forall i = 1,\dots,N \right\},$$

then we formally define a decision version of the problem ($\mathrm{P_{DD}}$) as follows. Without loss of generality we assume all data are integer.

Problem 1. Maximum weight b-matching over Permutation Group
Instance: *Undirect 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. The problem is clearly in NP because the problem $F(s,b)$ can be solved in a polynomial time [3]. We use a reduction from SAT for showing NP-completeness. For any instance of SAT, let $U = \{u_1, u_2, \dots, u_p\}$ and $C = \{c_1, c_2, \dots, c_q\}$ denote the set of variables and the set of clauses, respectively. We construct graph $G(V,E)$ and parameters as follows.

$$\begin{aligned}
V &= V_c \cup V_o \cup V_r \cup V_t \cup \left\{ \bigcup_{i=1,\dots,p} V_u^i \right\},\\
V_c &= \{v_1^c, v_2^c, \dots, v_q^c\},\\
V_o &= \{v_{1,1}^o, \dots v_{1,q}^o, v_{2,1}^o, \dots v_{2,q}^o, \dots, v_{q,1}^o, \dots v_{q,q}^o\},\\
V_u^1 &= \{v_1^u, v_1^{\neg u}\},\\
V_u^2 &= \{v_2^u, v_2^{\neg u}\},\\
&\vdots\\
V_u^p &= \{v_p^u, v_p^{\neg u}\},\\
V_r &= \{v_1^r, v_2^r, \dots, v_p^r\},\\
V_t &= \{v_{1,1}^t, \dots v_{1,q}^t, v_{2,1}^t, \dots v_{2,q}^t, \dots, v_{p,1}^t, \dots v_{p,q}^t\},\\
E &= E_{c,o} \cup E_{r,t} \cup E_{u,r} \cup E_{c,u} \cup E_{c,\neg u},\\
E_{c,o} &= \{\{v_i^c, v_{i,j}^o\} \mid i = 1,\dots,q, j = 1,\dots,q\},\\
E_{r,t} &= \{\{v_i^r, v_{i,j}^t\} \mid i = 1,\dots,p, j = 1,\dots,q\},\\
E_{u,r} &= \{\{v_i^u, v_i^r\} \mid i = 1,\dots,p\} \cup \{\{v_i^{\neg u}, v_i^r\} \mid i = 1,\dots,p\},\\
E_{c,u} &= \{\{v_i^c, v_j^u\} \mid j = 1,\dots,p, i = 1,\dots,q,\\
&\qquad \text{and clause } c_i \text{ contains variable } u_j\},\\
E_{c,\neg u} &= \{\{v_i^c, v_j^{\neg u}\} \mid j = 1,\dots,p, i = 1,\dots,q,\\
&\qquad \text{and clause } c_i \text{ contains variable } \neg u_j\},
\end{aligned}$$

$$s_e = \begin{cases} 1, & \text{if } e \in E_{c,u} \cup E_{c,\neg u}\\ M, & \text{if } e \in E_{c,o} \cup E_{r,t}\\ N, & \text{if } e \in E_{u,r} \end{cases}, \ \forall e \in E,$$

$$\hat{b}_i = \begin{cases} q+1, & \text{if } i \in V_c \cup V_o \cup V_r \cup V_t \cup \{v_1^u, \dots, v_p^u\}\\ 0, & \text{if } i \in \{v_1^{\neg u}, v_2^{\neg u}, \dots, v_p^{\neg u}\} \end{cases},$$

$\forall i \in V,$

$$L = (q^2+pq)M + pN + q,$$

where $N := 2pq+1$ and $M := 2pN+1$. Note that there can be at most $2pq$ edges with edge score 1 (i.e., $s_e = 1$) while exactly q^2+pq edges and $2p$ edges have edge scores M and N, respectively.

Claim 1.1. *For any permutation P, $F(s,P\hat{b}) \le L$ holds.*

Proof. Assume, for a contradiction, that for some permutation $\hat{P}$ we have $F(s,\hat{P}\hat{b}) > L$ with matching solution $\hat{x}$. It is obvious that $\hat{x}_e = 1$ for all $e \in E_{c,o} \cup E_{r,t}$ so that $(\hat{P}\hat{b})_i = q+1$ for all $i \in V_c \cup V_r$, that implies $\sum_{e\in E_{u,r}} \hat{x}_e = q$. Thus, $F(s,\hat{P}\hat{b}) = (q^2+pq)M + pN + \sum_{e\in E_{c,u}\cup E_{c,\neg u}} \hat{x}_e = L - q + \sum_{e\in E_{c,u}\cup E_{c,\neg u}} \hat{x}_e$. we have, by assumption, $\sum_{e\in E_{c,u}\cup E_{c,\neg u}} \hat{x}_e > q$ that means there exists $i^* \in V_c$ such that $(\hat{P}\hat{b})_{i^*} > q+1$ which derives a contradiction. □

Let $\mathbb{P}$ be the set of all permutation matrices. And let $\hat{\mathbb{P}} := \{P \in \mathbb{P} \mid (P\hat{b})_i = q+1$, for all $i \in V_o \cup V_c \cup V_r \cup V_t$, and $(P\hat{b})_{v_j^u} + (P\hat{b})_{v_j^{\neg u}} = q+1$ for all $j = 1,\dots,p\}$, i.e., $\hat{\mathbb{P}}$ is a set of perturbations that all nodes in $V_o \cup V_c \cup V_r \cup V_t$ have degree constraints of $q+1$ and exactly one of two nodes in V_u^j has node degree constraint of $q+1$.

Claim 1.2. *$P \in \hat{\mathbb{P}}$ if and only if $F(s,P\hat{b}) \ge L - q$.*

Proof. The sufficient condition is obvious. For showing the necessary condition, assume that there exists $P^* \in \mathbb{P}\backslash\hat{\mathbb{P}}$ such that $F(s,P^*\hat{b}) \ge L - q$. It is clear that $(P^*\hat{b})_i = q+1$ for all $i \in V_o \cup V_c \cup V_r \cup V_t$ (otherwise $F(s,P^*\hat{b}) \le L - M$). Since $P^* \in \mathbb{P}\backslash\hat{\mathbb{P}}$ there is some i^* such that $(P^*\hat{b})_{v_{i^*}^u} + (P^*\hat{b})_{v_{i^*}^{\neg u}} = 0$ that implies $F(s,P^*\hat{b}) \le L - N$ which derives a contradiction. □

Claim 1.3. *If $P \in \hat{\mathbb{P}}$, $F(s,P\hat{b}) = L - q + \sum_{e\in E_{c,u}\cup E_{c,\neg u}} x_e^*$, where x^* is a solution of b-matching problem $F(s,P\hat{b})$.*

Proof. This is clear by Claim 1.2. □

For any permutation $P \in \hat{\mathbb{P}}$, we define truth assignment $T_P : U \to \{true, false\}$ as follows: For all $i = 1,\dots,p$,

$$T_P(i) = \begin{cases} true, & \text{if } (P\hat{b})_{v_i^u} = q+1 \text{ and } (P\hat{b})_{v_i^{\neg u}} = 0;\\ false, & \text{if } (P\hat{b})_{v_i^u} = 0 \text{ and } (P\hat{b})_{v_i^{\neg u}} = q+1. \end{cases}.$$

We now show that C is satisfiable if and only if there is a permutation matrix $\tilde{P}$ such that $F(s,\tilde{P}\hat{b}) = L$.

For a sufficient condition, assume that C is satisfiable for truth assignment T^*. We consider a permutation matrix $P^* \in \hat{\mathbb{P}}$ corresponding truth assignment T^*. By Claim 1.1 and 1.2, it is clear that $L - q \le F(s,P^*\hat{b}) \le L$. Let x^* be the matching solution of b-matching problem $F(s,P^*\hat{b})$. For each clause i, we have $\sum_{e\in\{\{v_i^c,v_j^u\},\{v_i^c,v_j^{\neg u}\}\mid j=1,\dots,p\}} x_e^* = 1$,

because every clause in C is true and $(P^*\hat{b})_{v_i^c} = q+1$. By Claim 1.3, this implies $F(s, P^*\hat{b}) = L$.

For a necessary condition, assume that C is not satisfiable. We should show that for any $P \in \mathbb{P}$ we have $F(s, P\hat{b}) < L$. Assume that, for contradiction, there exists $\tilde{P}$ such that $F(s, \tilde{P}\hat{b}) = L$. By Claim 1.2, $\tilde{P} \in \hat{\mathbb{P}}$, and by Claim 1.3, we have $\sum_{e \in E_{c,u} \cup E_{c,\neg u}} \tilde{x}_e = q$ where $\tilde{x}$ is a solution of problem $F(s, \tilde{P}\hat{b})$. This means we have a truth assignment $T_{\tilde{P}}$ that satisfies every clause in C which derives a contradiction. This completes the proof.

4.2 Approximating of (P_{DD})

Since the problem (P_{DD}) is NP-hard, we develop the following approximating scheme.

For a given number K, the range of the node degrees was divided into K intervals, where a_k for all $k = 1, \ldots, K+1$ denote the dividing points. Let g_k denote the number of nodes having the node degrees belonging to interval k, which can be obtained by

$$g_k := \left[N \times \int_{a_k}^{a_{k+1}} \hat{\mathcal{P}}(z)dz \right], \tag{4}$$

where $[\cdot]$ is a function which returns the nearest integer. Introducing binary variables y_i^k, whose value is 1 if node i has node degree a_k, and 0 otherwise, the problem (P_{DD}^R) is obtained sa follows:

$$\text{maximize} \sum_{e \in E} s_e x_e - D \sum_{i \in V} s_i \tag{5}$$

$$\text{subject to} \sum_{e \in \sigma_i} x_e \leq \sum_{k=1,\ldots,K} a_k y_i^k + s_i, \ \forall i \in V, \tag{6}$$

$$\sum_{k=1,\ldots,K} y_i^k \leq 1, \ \forall i \in V, \tag{7}$$

$$\sum_{i \in V} y_i^k \leq g_k, \ \forall k = 1, \ldots, K, \tag{8}$$

$$x_e \in \{0,1\}, \ \forall e \in E, \tag{9}$$

$$y_i^k \in \{0,1\}, \ \forall k = 1, \ldots, K, i \in V, \tag{10}$$

$$s_i \geq 0, \ \forall i \in V. \tag{11}$$

The variables s_i for all $i \in V$ relaxes the node degree restriction, while the parameter D controls the degree of relaxation of the node degree distribution constraints. Constraints (7) ensure that no node can belong to more than one node degree interval.

We used a simple rounding heuristic: we solve the linear relaxation of (P_{DD}^R) and round off the (possibly) fractional solution x_e for all $e \in E$ to obtain an integer solution.

5. EXPERIMENTAL RESULTS

In this section, we report the computational results of the proposed algorithm. All algorithms were implemented using `Matlab`, and `R` was used only for the time–series analysis. The optimization problems were solved by `CPLEX`.

5.1 Facebook Networks

Due to the advance of internet technology, the social network like Facebook is becoming increasingly popular recently. Unlike the Enron e-mail network and stock correlation network, the Facebook friend network is an ever growing network. 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 [25]. From the original dataset that have 63.731 distinct individuals, we made two datasets—Facebook500 and Facebook1000—that contain the first 500 individuals for Facebook500 and 1000 individuals for Facebook1000 and links between only them. Not all of links in the dataset have the time of link establishment showing when the link was made. So, we first constructed a base network G_0 having links that do not have the time information. We then created networks for every two months having the newly created links only during that period, which results in 14 networks $(G_1, \ldots, G_{14})$ spanning from Sep. 2006 to Dec. 2008. The goal is to predict the newly associated friend links at time T from the information of networks $G_0, G_1, \ldots, G_{T-1}$. Let $\hat{E} \subseteq E$ denote the set of edges created before period T. The SO methods 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 setting $x_e = 1, \ \forall e \in \hat{E}$.

5.2 Baseline Methods

We aggregated all past networks $G_0 \sim G_{T-1}$ that represents the topology of the network just before the time of prediction, where the probability (score) matrix was calculated on.

Static Scoring Method (ST).

We built the static scores S_{ADA}, S_{KZ}, and S_{PA} using the scoring algorithms ADA [1], KZ [14], and PA [6, 20, 21], respectively, from the reduced graph of the past networks. Then, each scoring matrix was normalised by dividing it by the maximum score. The static score matrix S_{ST} can then be calculated as the average of all score matrices, i.e., $S_{ST} := (S_{ADA} + S_{KZ} + S_{PA})/3$.

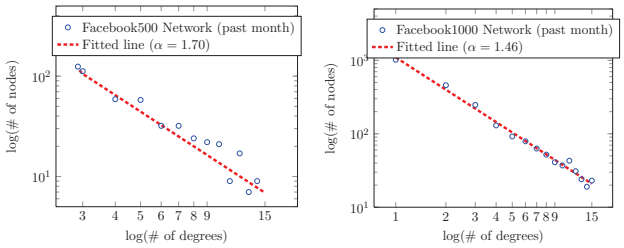
Hierarchical Random Graph Model (HRM).

We calculated the connection probability p_{ij} for every link in the network by using the hierarchical random graph model [9] by using the code provided at www.santafe.edu/∼aaronc/randomgraphs/. Let S_{HRM} denote the probability matrix whose (i, j) element represents the probability of link (i, j).

Hybrid Scoring Method (ALL:ST+HRM).

This method combines the static information and the link probability based on HRM by summing three score matrices; i.e., $S_{ALL} := (S_{ST} + S_{HRM})/2$.

For the performance measure of each algorithm, we used a receiver operation characteristics (ROC) curve [8] which 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). After obtaining the ROC curve, we calculate the area under curve (AUC) value from the plot. The AUC value 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. There are two methods of link prediction (SO and DD) and three scoring matrixes (ST, HRM, and ALL). We denote X_Y for the prediction made by prediction method X



(a) Facebook500 (Jan. 2008∼Feb. 2008) (b) Facebook1000 (Jan. 2008∼ Feb. 2008)

Figure 1: Examples of node degree distributions in the log-log plot.

and score matrix Y. Consequently, six algorithms were used in this study: SO_{ST}, SO_{HRM}, SO_{ALL}, DD_{ST}, DD_{HRM}, and DD_{ALL}.

5.3 Estimation of Node Degree Distribution

A power-law node degree distribution appears in the log-log plot as a straight line with a negative slope [24]. For each monthly network at period T, we solved the least-square linear fitting using the function

$$\log P(d) = C - \alpha \log d, \tag{12}$$

over the aggregated node degree histograms of the past networks. The value of α varies slightly over time due to environmental changes in the networks. Figure 1 shows examples of node degree distributions and the fitted lines for two tested networks.

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:

$$\hat{C} = \frac{1-\hat{\alpha}}{u^{1-\hat{\alpha}} - l^{1-\hat{\alpha}}}, \tag{13}$$

where l and u denote the minimum and the maximum degree values, respectively.

It should ne noted that without the scale-free property, it is not easy to determine the parameter of degree distribution. For example, in the case of a normal distribution, we should determine the mean and standard deviation of degree distribution. However, the parameters of degree distribution may change with network size, which means the distribution is not an invariant characteristic of the networks.

5.4 AUC Results for Facebook Networks

Table 1 and 2 summarize the performance of various algorithms, where the best AUC values are shown in bold face. The last row is for p-values of the paired and one-sided student t-test with the alternative hypothesis: *the average AUC of DD approach is better than the average of SO method*. The lower p-value is preferred. All scoring methods except HRM improved by our approach.

DD_{ALL} showed the best performance for the Facebook500 while DD_{ST} performed best for the Facebook1000 networks. The performance of SO_{ST} and SO_{ALL} was also greatly improved by the degree distributional approach. The HRM method did not perform well especially for the Facebook1000 because the HRM method relied on hierarchical decompositions (dendrograms) of the given network, of which size is growing exponentially with the network size.

Table 1: Results for the Facebook500 networks.

month-year	Static		HRM		All	
	SO_{ST}	DD_{ST}	SO_{HRM}	DD_{HRM}	SO_{ALL}	DD_{ALL}
9-2006∼10-2006	0.9336	0.9529	**0.9751**	0.9727	0.9638	0.9721
11-2006∼12-2006	0.9030	0.9268	0.8602	0.8607	0.9161	**0.9344**
1-2007∼2-2007	0.9118	**0.9314**	0.9161	0.9122	0.9186	0.9255
3-2007∼4-2007	0.9111	0.9392	0.9450	**0.9525**	0.9216	0.9432
5-2007∼6-2007	0.9139	0.9406	0.9202	0.9232	0.9378	**0.9435**
7-2007∼8-2007	0.9189	0.9471	0.8887	0.8867	0.9525	**0.9652**
9-2007∼10-2007	0.8335	0.8733	0.8285	0.8222	0.8574	**0.8762**
11-2007∼12-2007	0.8781	0.9023	0.8670	0.8696	0.9115	**0.9143**
1-2008∼2-2008	0.7744	0.7934	0.7220	0.6993	0.7978	**0.8026**
3-2008∼4-2008	0.8761	0.8963	0.8803	0.8789	0.9122	**0.9165**
5-2008∼6-2008	0.8433	0.8594	**0.8799**	0.8756	0.8620	0.8667
7-2008∼8-2008	0.8281	0.9138	0.8903	0.8984	0.8832	**0.9241**
9-2008∼10-2008	0.8249	0.8781	0.8312	0.8338	0.8586	**0.8784**
11-2008∼12-2008	0.8425	**0.8888**	0.8148	0.8160	0.8712	0.8832
average	0.8709	0.9031	0.8728	0.8716	0.8975	**0.9104**
p-value		0.0000		0.7291		0.0002

Table 2: Results for the Facebook1000 networks.

month-year	Static		HRM		All	
	SO_{ST}	DD_{ST}	SO_{HRM}	DD_{HRM}	SO_{ALL}	DD_{ALL}
9-2006∼10-2006	0.9205	**0.9421**	0.8481	0.8461	0.9169	0.9338
11-2006∼12-2006	0.8763	**0.9027**	0.8435	0.8432	0.8752	0.8980
1-2007∼2-2007	0.9265	**0.9478**	0.8827	0.8825	0.9184	0.9375
3-2007∼4-2007	0.8869	**0.9082**	0.7120	0.7026	0.8789	0.9049
5-2007∼6-2007	0.9132	**0.9373**	0.7992	0.8005	0.9103	0.9315
7-2007∼8-2007	0.8977	**0.9222**	0.8152	0.8140	0.8915	0.9106
9-2007∼10-2007	0.8446	**0.8859**	0.7572	0.7529	0.8446	0.8805
11-2007∼12-2007	0.8419	**0.8653**	0.7868	0.7761	0.8419	0.8616
1-2008∼2-2008	0.7476	**0.8180**	0.7310	0.7190	0.7476	0.7815
3-2008∼4-2008	0.8898	0.9223	0.7399	0.7405	0.8898	**0.9230**
5-2008∼6-2008	0.8341	**0.8924**	0.7799	0.7745	0.8341	0.8803
7-2008∼8-2008	0.7998	0.8671	0.7074	0.7111	0.7998	**0.8675**
9-2008∼10-2008	0.8228	**0.8740**	0.7138	0.7162	0.8228	0.8728
11-2008∼12-2008	0.7587	0.8131	0.6591	0.6616	0.7587	**0.8135**
average	0.8543	**0.8927**	0.7697	0.7672	0.8522	0.8855
p-value		0.0000		0.9534		0.0000

It should ne noted that our node degree restriction algorithm actually tends to suppress the prediction of high-scored links. For example, the simple ordering algorithm always produces better (or equal to) total sum of scores than our approach when the same number of links were predicted. However, it is clearly seen from the results that simply maximizing of the sum of scores does not necessarily yield a better prediction performance.

6. CONCLUSION

We propose a novel approach to the link prediction problem by exploiting a network-wide characteristic to improve prediction accuracy. Traditional link prediction algorithms are often based on the likelihood measure of each single link. These algorithms are relatively simple to implement and often perform well, however tthey fall short when the collective characteristics among many links are considered. More recently, a large number of studies have revealed that many real-world networks have the power-law of the node

degree distribution, which indicates that the network is scale free. We developed a mathematical programming formulation that makes the resulting link prediction solution follow the node degree distribution estimated from the pase networks.

We tested our algorithm using Facebook networks, where we were able to clearly demonstrate that each node degree distribution of each network actually 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. One of the most appealing features of our method is that it can be used in conjunction with any scoring method as presented in this study.

7. REFERENCES

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

[2] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDMÂąÂŕ06: Workshop on Link Analysis, Counter-terrorism and Security*. Citeseer, 2006.

[3] R. Anstee. A polynomial algorithm for b-matchings: an alternative approach. *Information Processing Letters*, 24(3):153–157, 1987.

[4] J. Bader, A. Chaudhuri, J. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature biotechnology*, 22(1):78–85, 2003.

[5] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.

[6] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica a-Statistical Mechanics and Its Applications*, 311(3-4):590–614, 2002.

[7] V. Boginski, S. Butenko, and P. Pardalos. Mining market data: a network approach. *Computers & Operations Research*, 33(11):3171–3184, 2006.

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

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

[10] W. Cook and W. Pulleyblank. Linear systems for constrained matching problems. *Mathematics of Operations Research*, 12(1):97–120, 1987.

[11] Z. Huang, X. Li, and H. Chen. Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 141–142. ACM, 2005.

[12] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[13] B. Karrer and M. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

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

[15] H. Kim, I. Kim, Y. Lee, and B. Kahng. Scale-free network in stock markets. *Journal of Korean Physical Society*, 40:1105–1108, 2002.

[16] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[17] L. Lu and T. Zhou. Link prediction in complex networks: A survey. *Arxiv preprint arXiv:1010.0725*, 2010.

[18] C. Manning, P. Raghavan, H. Schütze, and E. Corporation. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, UK, 2008.

[19] A. Medina, I. Matta, and J. Byers. On the origin of power laws in Internet topologies. *ACM SIGCOMM Computer Communication Review*, 30(2):18–28, 2000.

[20] M. E. J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[21] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409, 2001.

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

[23] G. Salton. Automatic text processing: the transformation. *Analysis and Retrieval of Information by Computer*, 1989.

[24] J. Shetty and J. Adibi. The Enron email dataset database schema and brief statistical report. *Information Sciences Institute Technical Report*, 2004.

[25] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[26] S. Zhou and R. Mondragón. Accurately modeling the Internet topology. *Physical Review E*, 70(6):066108, 2004.