Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 7213432, 11 pages
http://dx.doi.org/10.1155/2016/7213432

Research Article

Link Prediction via Sparse Gaussian Graphical Model

Liangliang Zhang, Longqi Yang, Guyu Hu, Zhisong Pan, and Zhen Li

College of Command Information System, PLA University of Science and Technology, Nanjing 210007, China

Correspondence should be addressed to Zhisong Pan; hotpzs@hotmail.com

Received 10 November 2015; Revised 27 January 2016; Accepted 27 January 2016

Academic Editor: David Bigaud

Link prediction is an important task in complex network analysis. Traditional link prediction methods are limited by network topology and lack of node property information, which makes predicting links challenging. In this study, we address link prediction using a sparse Gaussian graphical model and demonstrate its theoretical and practical effectiveness. In theory, link prediction is executed by estimating the inverse covariance matrix of samples to overcome information limits. The proposed method was evaluated with four small and four large real-world datasets. The experimental results show that the area under the curve (AUC) value obtained by the proposed method improved by an average of 3% and 12.5% compared to 13 mainstream similarity methods, respectively. This method outperforms the baseline method, and the prediction accuracy is superior to mainstream methods when using only 80% of the training set. The method also provides significantly higher AUC values when using only 60% in Dolphin and Taro datasets. Furthermore, the error rate of the proposed method demonstrates superior performance with all datasets compared to mainstream methods.

1. Introduction

Since 2010, link prediction has become an increasingly distinctive and important part of complex network analysis. Link prediction refers to the prediction of a possible link between two nodes when links are unknown [1]. Such prediction involves the prediction of existing yet unknown links and future links. Link prediction is the basis of data mining problems and lays the foundation for complex network research. Link prediction provides a mechanism for both structure and evolution of networks. Studying this problem is important from both theoretical and practical perspectives [2]. Existing community detection research is primarily based on an adjacency matrix, and community detection typically depends on the adequacy and completeness of the adjacency matrix. Link prediction is instrumental for accurately analysing social-network structures, helping community detection, and improving the accuracy of community detection [2, 3]. Link prediction can be used to predict missing data and can help analyse network evolution [4]. For example, we can use the current network structure to predict users who have not been recognized as friends or can develop into friends.

Link prediction methods have made remarkable achievements in various fields, including biology, social science, security, and medicine. Ermiş et al. [2] address the link prediction problem by data fusion formulated as simultaneous factorization of several observation tensors where latent factors are shared among each observation; some studies turn to multirelational link prediction [3]; Yang et al. also studied evaluation of link prediction [5].

Gong et al. in 2014 extended the Social-Attribute Network framework with several supervised and unsupervised link-prediction algorithms and demonstrate their method performance improvement [6]. However, such methods have limitations when processing social datasets. First, social datasets are low-quality datasets that include faulty links and noise. Such datasets must be preprocessed before similarity measurement, set partitioning, and common neighbour (CN) count. Moreover, node properties are cumbersome to obtain, and most social network data can only be used to obtain a raw adjacency matrix that does not include specific attribute information because user information is private in most online systems. Consequently, many prediction methods cannot use the features of such properties, and we cannot calculate feature property. Therefore, using only an adjacency matrix can avoid interference of node properties, which is convenient and feasible.

Many community detection methods are based on an adjacency matrix. Thus, the adjacency matrixes integrity directly affects the results. Through link prediction using an adjacency matrix, we can determine the relationships between unconnected nodes, and the entire community structure can be obtained by analysing these relationships. Thus, an effective link prediction method is required. This issue presents a series of challenges such as the following: (1) link prediction must function with an adjacency matrix that does not contain properties, (2) a graph structure model for estimating the network is required to determine the role of different types of connections, and (3) verification and evaluation of link prediction are required.

Existing link prediction methods use similarities, node properties, edge properties, and so forth. However, properties require a large amount of test data and heavily rely on network connectivity and structure; thus, link prediction without properties is less robust. When the network structure changes, it is difficult to mine the relationships between nodes. Thus, determining how to use limited test data to predict a network edge is the motivation of this study.

To solve the above problems, this paper presents a link prediction method based on the application of a Gaussian graphical model (GGM) to an adjacency matrix. This concept references Friedman et al.'s [7] sparse inverse covariance estimation theory. The study uses the original adjacency matrix for sampling, thereby obtaining a sample matrix. Thus, we use a sparse GGM (SGGM) inverse covariance matrix for link prediction. The main contributions of this study are as follows:

(1) Sampling of a network to build a feature matrix, seeking maximum likelihood estimation using an SGGM and estimating an inverse covariance matrix (precision matrix) of the adjacency matrix.

(2) Establishing conditional independence between nodes to complete link prediction using the Markov random field independence principle.

(3) Proving that the proposed method is more effective than previous methods by testing the methods using four real-world datasets.

The remainder of this paper is organised as follows. We introduce related work in Section 2, including many previous link prediction methods. In Section 3, we present our SGGM-based link prediction method. In Section 4, we introduce eight real-world datasets and test the methods using these datasets to prove that the proposed method is more effective than previous methods. Finally, we conclude the paper and present suggestions for future work in Section 5.

2. Related Work

2.1. Problem Description. Existing link prediction methods can be divided into three categories.

2.1.1. Similarity Link Prediction Employs Different Methods. One is based on node properties, such as sex, age, occupation, preferences, and other properties, to compute node similarity. It is more probable that edges will exist between high-similarity nodes. Another method is based on the network structures similarity, for example, the use of CN nodes. However, this method is only applicable to a network with a high network clustering coefficient.

2.1.2. Estimates Based on the Maximum Likelihood Estimation of a Link Can Be Divided into Two Categories. One method is based on network hierarchy, but it has high complexity because it generates many network samples. The other method is based on stochastic block model prediction, wherein nodes are divided into some sets and the probability of an edge depends on corresponding sets.

2.1.3. A Link Prediction Model Based on Probability Builds a Model by Adjusting the Parameters. This can fit the structure of the relationships in real networks. A pair of nodes will generate an edge determined by probability using the optimum parameter. A probabilistic model considers the probability of existing edges as a property. It transforms edge prediction into property issues. This method takes advantage of the network structure and node properties with high precision but offers poor universality.

Due to the poor universality of maximum likelihood estimation and the probability model, which depend highly on node properties, these methods cannot be applied to many networks. Herein, we consider a link prediction method which is only based on similarity and discuss experiments performed to compare the proposed and previous methods.

2.2. Similarity-Based Link Prediction. Here, we compare the prediction accuracies of 13 similarity measures. All of these measures are based on the local structural information contained in a test set. We first introduce each measure briefly. The formulas are shown in Table 1. Here, $G(V,E)$ is an undirected network, V is a set of nodes, and E is a set of edges. The total number of nodes for the network is N and the number of edges is M. For a node X and its neighbours $\Lambda(x)$, the degree of X is $d(x) = |\Lambda(x)|$. The network has $N(N-1)/2$ node pairs, that is, a universal set U. When given a link prediction method, each pair of nodes $(x,y) \in (U \setminus E)$ without an edge will have a score S_{xy}. Then, all unconnected pairs of nodes are ordered by the score value in descending order and the probability of an edge appearing is the largest on the top.

CN nodes are based on a local information similarity index, and this is one of the simplest methods [8]. In other words, it is more likely that a link will exist between two high-similarity nodes that have many neighbours. For node X, $\Lambda(x)$ represent its neighbours, and if nodes X and Y have many CNs, they are more likely to have a link. Visibly, the structural equivalence pays more attention to whether two nodes are in the same circumstance. For example, in a social network context, if two people share many friends, they are more likely to be friends themselves. We consider

Table 1: Similarity indexes of local node information.

Name	Definition
CN	$S_{xy} = \lvert \Lambda(x) \cap \Lambda(y) \rvert$
Salton index	$S_{xy} = \dfrac{\lvert \Lambda(x) \cap \Lambda(y) \rvert}{\sqrt{d(x) \times d(y)}}$
Jaccard index	$S_{xy} = \dfrac{\lvert \Lambda(x) \cap \Lambda(y) \rvert}{\lvert \Lambda(x) \cup \Lambda(y) \rvert}$
Sørensen index	$S_{xy} = \dfrac{2 \lvert \Lambda(x) \cap \Lambda(y) \rvert}{d(x) + d(y)}$
HPI	$S_{xy} = \dfrac{\lvert \Lambda(x) \cap \Lambda(y) \rvert}{\min \{d(x), d(y)\}}$
HDI	$S_{xy} = \dfrac{\lvert \Lambda(x) \cap \Lambda(y) \rvert}{\max \{d(x), d(y)\}}$
LHN-I	$S_{xy} = \dfrac{\lvert \Lambda(x) \cap \Lambda(y) \rvert}{d(x) \times d(y)}$
PA	$S_{xy} = d(x) \times d(y)$
AA	$S_{xy} = \sum_{z \in \Lambda(x) \cap \Lambda(y)} \dfrac{1}{\lg d(z)}$
RA	$S_{xy} = \sum_{z \in \Lambda(x) \cap \Lambda(y)} \dfrac{1}{d(z)}$

the impact of degree of both nodes and generate six similarity indices, namely, Salton index (cosine-based similarity) [9], Jaccard index [10], Sørensen index [11], hub promoted index (HPI), hub depressed index (HDI), and Leicht-Holme-Newman (LHN) index [12]. These indices are based on CN similarity. Another similarity-degree-based method is preferential attachment (PA) [13], which can generate a scale-free network. Note that the complexity of this algorithm is lower than that of others because it requires less information.

If we consider degree information of two nodes CNs, we have the Adamic-Adar (AA) index [14], which considers that the contribution of a small degree of CN nodes is greater than that of a larger one. Liu et al. [15] proposed the RA index, which forms a view of network resource allocation. The RA and AA indices determine the weight of CN nodes in different ways; RA decreases by $1/k$ and AA is $1/\lg k$. The RA index performs better than the AA index in a weighted network and community mining. Traditional CN methods do not distinguish the different roles of CNs. Liu et al. [15] proposed a local naive Bayes (LNB) model. This model creates a role parameter for marking the different roles of CNs. However, it is only applied to certain types of networks.

3. Link Prediction Based on Sparse Gaussian Graphical Model

Most similarity algorithms perform well with ideal datasets and large training datasets. Such algorithms do not consider missing, incomplete, and polluted data in an adjacency matrix. As previously mentioned, the accuracy of similarity-based methods depends on whether the method can determine the characteristics of the network structure. For example, CN-based algorithms utilise a nodes CNs, and a pair of nodes with many CNs are more likely to connect. Such algorithms perform well, sometimes even better than complex algorithms, in a network with a high clustering coefficient. However, in networks with a low clustering coefficient, such as router or power networks, accuracy is significantly worse.

This study attempts to determine links in an adjacency matrix to reveal actual links between nodes. The proposed SGGM method is based on an undirected graphical model and transforms the adjacency matrix without using node properties. The SGGM method estimates the precision of the matrix of a network to predict unknown edges. To verify link prediction accuracy, area under the curve (AUC) and an error metric are used to prove the proposed methods' effectiveness.

3.1. Sparse Graphical Model. Biologists interested in genetic connections use the GGM to estimate genetic interaction. Edge relationships in an undirected graph are represented by the joint distribution of random variables. For example, genome work is based on biological functions, and some supervising relationships exist between genes. Corresponding to the graph, genes represent nodes and edges represent this supervising relationship. The relationship between genes provides a method to model such relationships. We assume that variables have Gaussian distribution; therefore, the GGM is most frequently used. Therefore, the problem is equivalent to estimating the inverse covariance matrix Σ^{-1} (precision matrix $\Theta = \Sigma^{-1}$), and the diagonal elements of precision matrix Θ represent the edges in the graph [16]. The GGM denotes the statistical dependencies between variables. If there is no link between two nodes, such nodes have conditional independence. In the GGM, a precision matrix can parameterise each edge.

A popular modern method for evaluating a GGM is the graphical lasso, and Friedman et al. [7] added an ℓ_1 norm to punish each off-diagonal element of the inverse covariance matrix.

The GGM encodes the conditional dependence relationships among a set of p variables and n observations that have identical and independent Gaussian distribution. Motivated by network terminology, we can refer to the p variables in a graphical model as nodes. If a pair of variables (or features) is conditionally dependent, then there is an edge between the corresponding pair of nodes; otherwise, no edge is present. The basic model for continuous data assumes that the observations have a multivariate Gaussian distribution with mean μ and covariance matrix Σ, and $X[1], X[2], \dots, X[N] \sim N(0, \Sigma)$. If the ijth component of Σ^{-1} is zero, then the variables i and j are conditionally independent given the other variables. Specifically, given X_k and $k = \{1, \dots, p\} \backslash \{i, j\}$, the nonzero elements in Σ^{-1} represent the graphs structure. In other words, if $(\Sigma^{-1})_{i,j} = 0$, nodes i and j have no connecting edge in the graph. The precision matrix is sparse due to the conditional independence of the variables. To estimate a sparse GGM, many methods are based on maximum likelihood estimation (MLE), and one such method is the graphical lasso, which is expressed as

$$\max_{\Theta \succ 0} \{\log \det \Theta - \operatorname{tr}(S\Theta) - \lambda \|\Theta\|_1\}, \tag{1}$$

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})^T, \tag{2}$$

$$\|\Theta\|_1 = \sum_{i \neq j} |\Theta_{ij}|. \tag{3}$$

Here, S is the experiential covariance matrix and λ is positive tuning parameter. If $\Theta > 0$, Θ is a $p \times p$ positive definite matrix. $\|\Theta\|_1$ is the punishment element, and Θ becomes increasingly sparse with increasing λ. An ℓ_1-norm is utilised to consider Θ a variable rather than a fixed parameter.

3.2. Graphical Lasso Algorithm. In 2008, Banerjee et al. [17] proved that formula (1) is convex. They estimated Σ via W only and subsequently solved the problem by optimising over each row. Moreover, they optimised the corresponding column of W in a block coordinate descent fashion:

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & s_{12} \\ s_{12}^T & s_{22} \end{pmatrix}. \tag{4}$$

By partitioning W and S, the solution for w_{12} satisfies

$$w_{12} = \arg\min_y \{y^T W_{11}^{-1} y : \|y - s_{12}\|_\infty \leq p\}. \tag{5}$$

Banerjee et al. [17] proved that solving (5) is equivalent to solving its dual problem:

$$\min_\beta = \left\{\frac{1}{2}\left\|W_{11}^{1/2}\beta - b\right\|^2 + \rho\|\beta\|_1\right\}. \tag{6}$$

Here, assume that β is the solution to (6); that is, $b = W_{11}^{1/2}s_{12}$; thus, $w_{12} = W_{11}\beta$ is the solution to (5). Therefore, (6) is similar to lasso regression and is the basis for the graphical lasso algorithm. First, we must prove that (6) and (1) are equivalent. Given $W\Theta = I$, that is,

$$\begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix}\begin{pmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^T & \theta_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0^T & 1 \end{pmatrix}, \tag{7}$$

the MLE of (1) is rewritten as

$$W - S - \rho H = 0. \tag{8}$$

Note that the derivative of $\log\det\Theta$ equals $\Theta^{-1} = W$. Here, $H_{ij} \in \operatorname{sign}(\Theta_{ij})$; that is, $H_{ij} = \operatorname{sign}(\Theta_{ij})$, if $\Theta_{ij} \neq 0$; else $H_{ij} \in [-1, 1]$ if $\Theta_{ij} = 0$. Thus, the upper-right block of (8) is expressed as

$$w_{12} - s_{12} - \rho\gamma_{12} = 0. \tag{9}$$

The subgradient equation of (6) is expressed as follows:

$$W_{11}\beta - s_{12} + \rho v = 0, \tag{10}$$

where $v \in \operatorname{sign}(\beta)$. Here, suppose that (W, H) solves (8); consequently, (w_{12}, γ_{12}) solves (9). Then, $\beta = W_{11}^{-1}w_{12}$ and $v = -\gamma_{12}$ solve (10). For the sign terms, $W_{11}\Theta_{12} + w_{12}\Theta_{22} = 0$ is derived from (7); therefore, we obtain $\Theta_{12} = -\Theta_{22}W_{11}^{-1}w_{12}$. Since $\Theta_{22} > 0$, it follows that $\operatorname{sign}(\Theta_{12}) = -\operatorname{sign}(W_{11}^{-1}w_{12}) = -\operatorname{sign}(\beta)$, which proves the equivalence. The solution β to the lasso problem (6) gives the relevant part of $\Theta_{12} = -\Theta_{22}\beta$.

Problem (6) appears similar to a lasso (ℓ_1-regularised) least-squares problem. In fact, if $W_{11} = S_{11}$, the solutions of β are equal to the lasso estimates for the pth variable. In 2007, Friedman et al. used fast coordinate descent algorithms to solve the lasso problem. Hsieh et al. [18, 19] proposed a novel block-coordinate descent method via clustering to solve the optimization problem. The method can reach quadratic convergence rates, which is designed for large network.

3.3. Link Prediction Scheme Based on Sparse Inverse Covariance Estimation. Here, we consider an undirected simple network $G(V, E)$, where E is the set of links and V is the set of nodes. Moreover, A is an adjacent matrix, Λ is the graphical lasso parameter, p denotes the number of nodes, and the universal set U consists of $p(p-1)/2$ edges. The set of links are randomly divided into two parts: training (E_T) and testing (E_P) sets. Note that only the information in the training set can be used for link prediction. ρ denotes the ratio of the training set which means the amount of edges being used. Clearly, $E = E_T \cup E_P$, and $E_T \cap E_P = \emptyset$. In the training set sampling process, we generated n independent observations $X = [x_1, \ldots, x_n]$, each from normal distribution $N(0, \Sigma)$, where $\Sigma = (E_T)^{-1}$. Here, S denotes the sample covariance matrices of $X = [x_1, \ldots, x_n]$. We then apply SGGM for link prediction. The pseudocode of the link prediction scheme is given in Algorithm 1.

4. Experiment and Analysis

4.1. Evaluation Metrics

4.1.1. AUC. In link prediction, we focus on accurately recovering missing links. Here, we have chosen the area under a relative operating characteristic (ROC) curve as a metric. An ROC curve represents a comparison of two operating characteristics TPR and FPR as the criterion changes. The AUC ranges from 0.5 to 1, and a higher value means a better model.

4.1.2. Error Rate. We also defined an error metric to evaluate the difference between the original and estimated networks and are denoted by θ and θ', respectively. The error metric is defined as Error $= \|\theta - \theta'\|_F / \|\theta\|_F$, where $\|\cdot\|_F$ is the Frobenius norm.

4.2. Real-World Datasets. In this paper, we consider four representative networks drawn from disparate fields (data

```
Require: $A, \rho > 0, \lambda > 0, n > 0$
Ensure: $\overline{A}$
    $E^T$: randomly select $\rho|E|$ edges from $E$.
    $E^P \Leftarrow E - E^T$
    $\Sigma \Leftarrow (E^T)^{-1}$
    $X \Leftarrow [x_1, x_2, \ldots, x_n]$: $n$ independent observations from $N(0, \Sigma)$
    $S \Leftarrow \mathrm{cov}(X)$
    $\Theta \Leftarrow$ Sparse Inverse Covariance Estimation ($A$, $\lambda$, $S$)
    if $\theta_{ij} \leq 0$ and $i \neq j$ then
        $\overline{A}_{ij} \Leftarrow 1$
    else
        $\overline{A}_{ij} \Leftarrow 0$
    end if
```

Algorithm 1: Link prediction based on sparse inverse covariance estimation with graphical lasso.

Table 2: Basic topological features of four real-world networks.

Network	N	M	E	C	Avg_D	r
Taro	22	39	0.488	0.339	3.546	−0.375
Tailor Shop	39	158	0.567	0.458	8.103	−0.183
Dolphin	62	159	0.379	0.259	5.129	−0.044
Football	115	613	0.450	0.403	10.661	0.162

sources: (1) http://www-personal.umich.edu/~mejn/netdata/ (Football and Dolphin) and (2) http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm (Tailor Shop and Taro)).

We selected four real-world datasets to compare the proposed method with the 13 similarity measures. Table 2 shows the basic topological features of the four real-world networks, wherein N and M are the total numbers of nodes and links, respectively. E is the network efficiency [20] which is defined as $E = (2/N(N-1)) \sum_{x,y \in V, x \neq y} d_{xy}^{-1}$, where d_{xy} denotes the shortest distance between nodes x and y and $d_{xy} = +\infty$ if x and y are in two different components. C and Avg_D denote the clustering coefficient [21] and average degree, respectively, and r is the network assortative coefficient [22]. The network is assortative if a node tends to connect with the approximate node. $r > 0$ means that the entire network has assortative structure, and a node with large degree tends to connect to other nodes with large degree. $r < 0$ means that the entire network has disassortative structure, and $r = 0$ implies that there is no correlation in the network structure. Figure 1 shows the distribution of the four datasets. It is clear that the degree distributions of the four data sets are considerably different.

Table 3: Football dataset results.

Method	Training set ratio			
	60%	70%	80%	90%
SGGM, $n = 0.5p$	0.702	0.725	0.757	0.783
SGGM, $n = p$	0.742	0.777	0.818	**0.857**
SGGM, $n = 1.5p$	0.753	0.795	**0.845**	**0.889**
SGGM, $n = 2p$	0.756	0.800	**0.852**	**0.905**
CN	0.774	0.798	0.824	0.835
Salton	**0.779**	**0.804**	0.828	0.838
Jaccard	0.502	0.508	0.514	0.520
Sørensen	0.590	0.597	0.604	0.606
HPI	0.778	0.802	0.827	0.838
HDI	**0.779**	0.803	0.827	0.838
LHN	0.777	0.802	0.827	0.838
AA	0.774	0.798	0.825	0.836
RA	0.774	0.798	0.824	0.836
PA	0.515	0.516	0.520	0.528
LNBCN	0.777	0.800	0.825	0.836
LNBAA	0.777	0.800	0.825	0.836
LNBRA	0.776	0.799	0.825	0.837

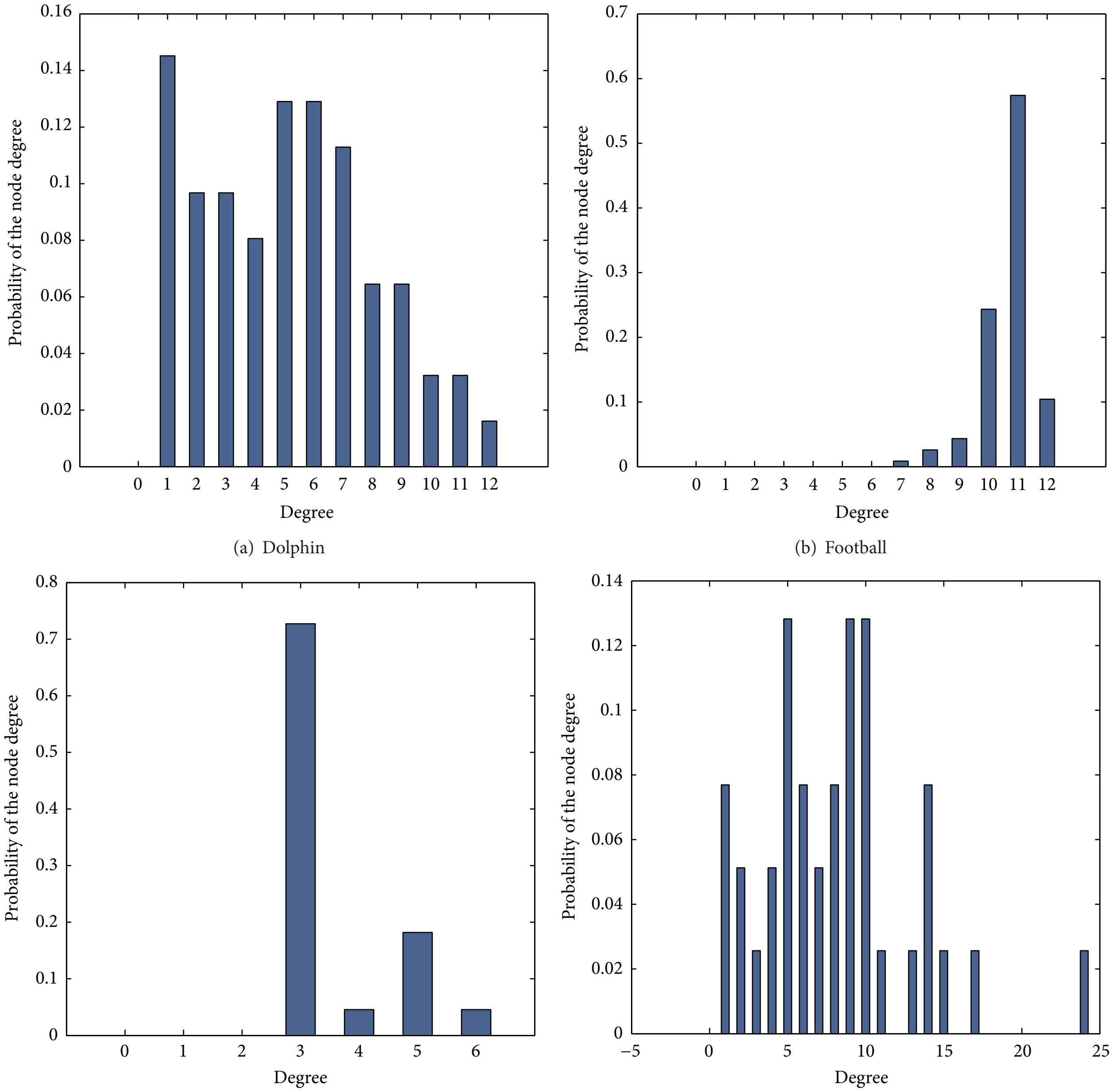
4.3. Experimental Settings. To evaluate the performance of the algorithms with different sized training sets, we randomly selected 60%, 70%, 80%, and 90% of the links from the original network as training sets. Only the information in the training set was used for estimation. Each training set was executed 10 times. We choose 10-fold cross validation because it has been widely accepted in machine learning and data mining research. Thirteen similarity measures were implemented following Zhou et al. [14], and the SGGM was implemented using the SLEP toolbox [23]. We evaluated the performance of the GGM algorithm with different sample scales. Here, p denotes the number of nodes. We set the sample scale to $0.5p$, p, $1.5p$, and $2p$. The parameter λ in the SGGM controlled the sparsity of the estimated model. Larger λ gives a sparser estimated network. Here, λ ranged from 0.01 to 0.2 with a step of 0.01.

4.4. Results

4.4.1. Tuning Parameter for the SGGM. Using the training set that was scaled to 90%, we tested the proposed SGGM with different λ and sample scales. As shown in Figure 2, the AUC of the SGGM increased with increasing sample scale for the four datasets. One advantage of the SGGM is that it is not sensitive to the parameter λ. Then, we set the sample scale to $0.5N$. Figure 3 shows that the AUC increased with increase in training set scale. For the Tailor Shop dataset, the proposed algorithm performs optimally with the 70% scaled training set.

4.4.2. Comparison on AUC. Table 3 lists the results for 14 methods with the Football dataset. The SGGM performs optimally when 80% or 90% of the training set is used and the number of samples is greater than N. The prediction accuracy of the SGGM increased by 5% compared to the other 13 methods. However, with lower proportions of the training set, the SGGMs AUC is very close to that of the other methods. Note that PA yielded the poorest result with this dataset.

As shown in Table 4, the SGGM method performs optimally with the Dolphin dataset. The prediction accuracy of the SGGM method increased by at least 3% compared to the



(a) Dolphin

(b) Football

(c) Taro

(d) Tailor Shop

Figure 1: Degree distribution of four real-world datasets.

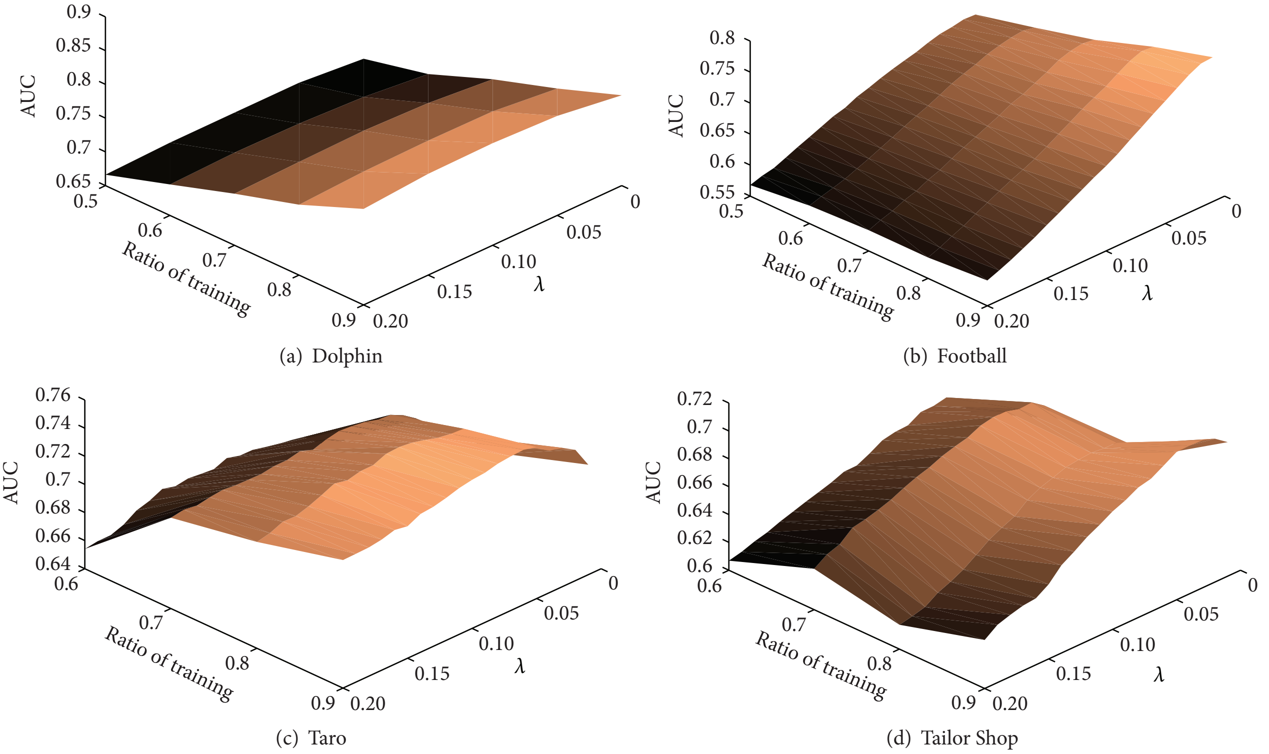
other thirteen methods. PA is optimal among the similarity measures, and Jaccard yielded the poorest result.

As shown in Table 5, the SGGM method outperforms the other 13 methods with the Taro dataset. The AUC improved by at least 10%. For the Taro dataset, the degree of 70% nodes is 3, which indicates a low degree of heterogeneity; thus, the performance of the other thirteen methods is very close. One might expect PA to show good performance on assortative networks and poor performance on disassortative networks. However, the experimental results show no obvious correlation between PA performance and the assortative coefficient.

For the Tailor Shop dataset, PA performs optimally when the 60% and 70% training sets were used, while the SGGM method performs optimally with the 80% and 90% training sets (Table 6). The SGGMs prediction accuracy gradually increased with increasing the number of samples. Thus, when sampling conditions permit, multiple samples could improve prediction accuracy.

4.4.3. Comparison on ROC. The ratio of the training set was set to 90%. The sample scale of the SGGM varied within $0.5p$, p, $1.5p$, and $2p$; $\lambda = 0.01$. CN was chosen as the representative neighbourhood because six other measures (Salton, Jaccard, Sørensen, HPI, HDI, and LHN) are variants of CN.

The ROC of the RA index is similar to that of the AA index. As observed in Figure 4, in most cases, the SGGM



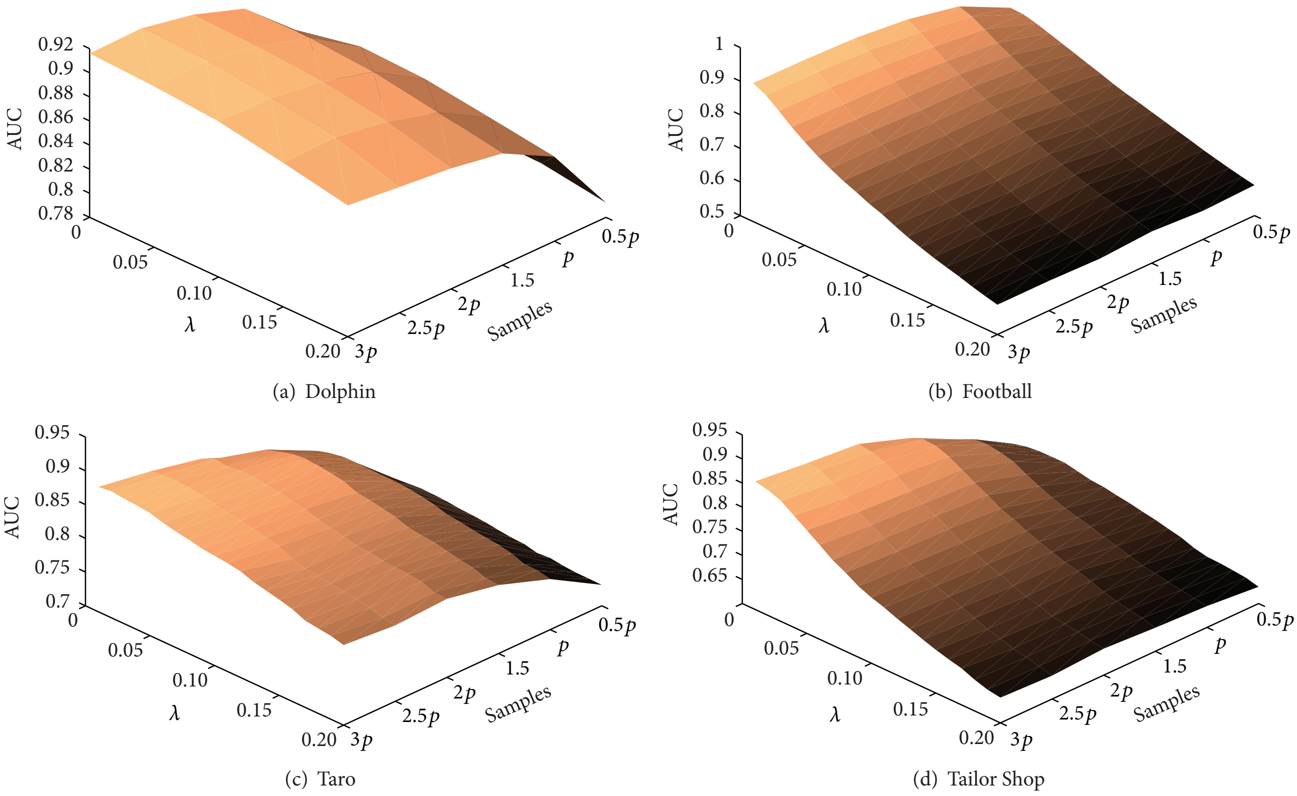
(a) Dolphin

(b) Football

(c) Taro

(d) Tailor Shop

Figure 2: AUC of the SGGM (four datasets, training set ratio = 90%).



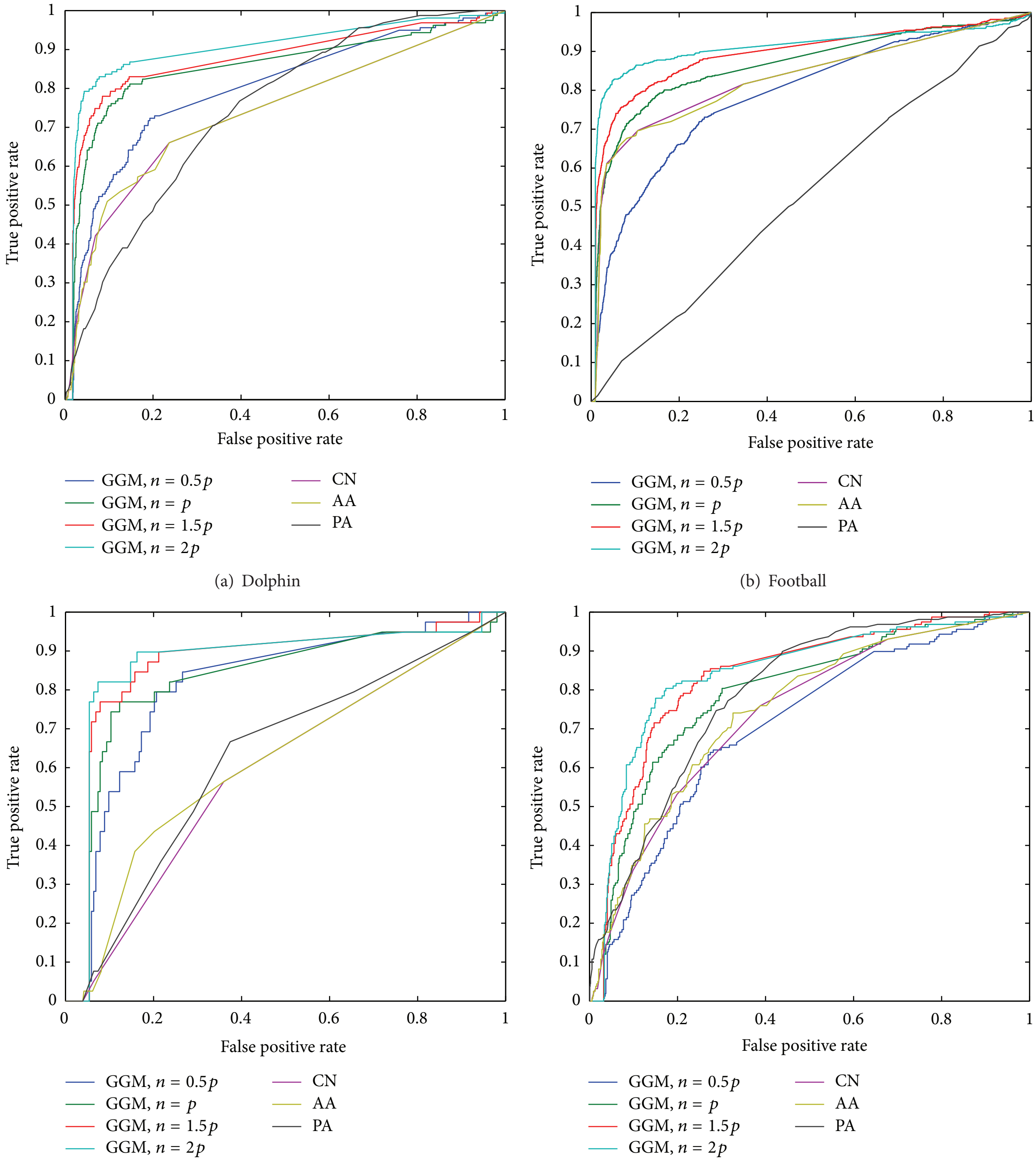
(a) Dolphin

(b) Football

(c) Taro

(d) Tailor Shop

Figure 3: AUC of the SGGM (four datasets, sample scale = 0.5N).



(a) Dolphin

(b) Football

(c) Taro

(d) Tailor Shop

FIGURE 4: Comparison of ROC metric of four methods (SGGM, CN, AA, and PA) on four datasets.

shows an improvement over existing similarity measures when sufficient samples were used. Note that more samples lead to higher accuracy.

4.4.4. Comparison on Error Rate. The error rate is a metric to evaluate the difference between two matrices. A lower error rate indicates that the estimated matrix approximates the original matrix. From Table 7, we observe that the SGGM has the lowest error rate among all methods shown. To show the effectiveness of the proposed SGGM, both the original adjacency matrix and estimated adjacency matrix are presented as a coloured graph in Figure 5. We used the Taro dataset with a 90% training set to recover the original network. Graphs with fewer red points indicate

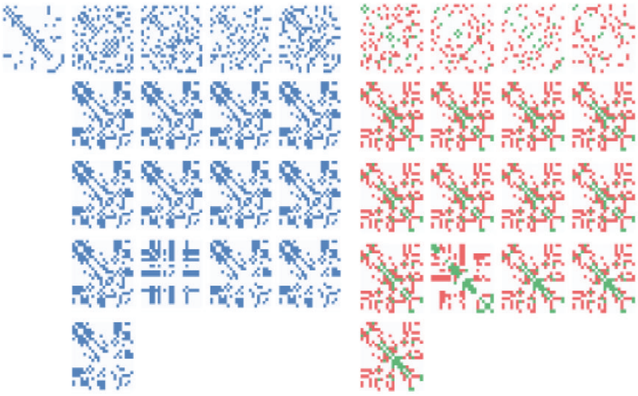


Figure 5: Recovery performances of 14 methods on the Taro dataset. The upper left corner is the original matrix; blue points denote links between two corresponding nodes. The red points denote links in the original matrix that were not accurately estimated. The green points denote links in the estimated matrix that do not exist in the original matrix. From left to right, top to bottom, the figures are the recovered matrix by SGGM ($n = 0.5p$, p, $1.5p$, and $2p$), CN, Salton, Jaccard, Sørensen, HPI, HDI, LHN, AA, RA, PA, LNBCN, LNBAA, and LNBRA, respectively.

Table 4: Dolphin dataset results.

Method	Training set ratio			
	60%	70%	80%	90%
SGGM, $n = 0.5p$	0.699	**0.731**	**0.761**	**0.801**
SGGM, $n = p$	**0.742**	**0.775**	**0.823**	**0.867**
SGGM, $n = 1.5p$	**0.758**	**0.803**	**0.845**	**0.895**
SGGM, $n = 2p$	**0.765**	**0.807**	**0.857**	**0.909**
CN	0.631	0.687	0.728	0.762
Salton	0.621	0.673	0.709	0.740
Jaccard	0.485	0.490	0.495	0.500
Sørensen	0.538	0.558	0.571	0.583
HPI	0.620	0.669	0.702	0.728
HDI	0.626	0.680	0.720	0.753
LHN	0.619	0.666	0.698	0.721
AA	0.633	0.690	0.732	0.767
RA	0.632	0.690	0.731	0.766
PA	0.712	0.727	0.737	0.746
LNBCN	0.636	0.696	0.739	0.775
LNBAA	0.635	0.694	0.738	0.773
LNBRA	0.635	0.693	0.736	0.771

Table 5: Taro dataset results.

Method	Training set ratio			
	60%	70%	80%	90%
SGGM, $n = 0.5p$	**0.672**	**0.711**	**0.736**	**0.750**
SGGM, $n = p$	**0.713**	**0.747**	**0.779**	**0.815**
SGGM, $n = 1.5p$	**0.721**	**0.763**	**0.816**	**0.857**
SGGM, $n = 2p$	**0.723**	**0.788**	**0.825**	**0.865**
CN	0.529	0.545	0.573	0.582
Salton	0.530	0.542	0.571	0.580
Jaccard	0.468	0.476	0.476	0.478
Sørensen	0.523	0.533	0.544	0.552
HPI	0.533	0.545	0.574	0.585
HDI	0.528	0.541	0.568	0.576
LHN	0.532	0.543	0.570	0.579
AA	0.533	0.554	0.589	0.609
RA	0.533	0.554	0.589	0.610
PA	0.559	0.568	0.582	0.607
LNBCN	0.536	0.562	0.616	0.638
LNBAA	0.537	0.561	0.614	0.634
LNBRA	0.536	0.560	0.613	0.634

Table 6: Tailor Shop dataset results.

Method	Training set ratio			
	60%	70%	80%	90%
SGGM, $n = 0.5p$	0.652	0.679	0.677	0.706
SGGM, $n = p$	0.703	0.723	0.750	0.769
SGGM, $n = 1.5p$	0.715	0.755	0.786	**0.817**
SGGM, $n = 2p$	0.734	0.770	**0.803**	**0.844**
CN	0.673	0.699	0.720	0.746
Salton	0.616	0.628	0.646	0.667
Jaccard	0.494	0.498	0.504	0.514
Sørensen	0.583	0.589	0.592	0.597
HPI	0.615	0.618	0.631	0.642
HDI	0.628	0.643	0.660	0.680
LHN	0.588	0.574	0.571	0.566
AA	0.682	0.709	0.729	0.753
RA	0.681	0.707	0.729	0.752
PA	**0.761**	**0.775**	0.779	0.786
LNBCN	0.692	0.715	0.746	0.765
LNBAA	0.693	0.716	0.746	0.764
LNBRA	0.693	0.715	0.745	0.762

good recovery of the original matrix. As can be seen from Figure 5, using the SGGM method can restore the original image, whereas the other similarity measures return greater error rates. Note that the SGGM error rate decreased with increasing sample scale.

4.5. Large Datasets. The main challenge of link prediction is dealing with large network. We carefully choose four large datasets from four individual domain. For SGGM method, λ is set to 0.01; 90% of data was used as training set.

In order to prove the performance of SGGM, we compare the proposed method with the other thirteen methods on these large datasets (for details see Table 8) in terms of AUC and error rate (data sources: (1) http://konect.uni-koblenz.de/networks/ (Email). (2) http://www3.nd.edu/~networks/resources.htm (Protein). (3) http://www-personal.umich.edu/~mejn/netdata/power.zip (Grid). (4) http://konect.uni-koblenz.de/networks/ (Astro-ph)).

To estimate the large network efficiently, we used QUIC [19] to solve sparse inverse covariance estimation (formula (1)) instead of glasso [7]. Overall, the proposed method

Table 7: Error rate of 13 methods on 4 datasets.

Method	Datasets			
	Football	Dolphin	Taro	Tailor Shop
SGGM, $n = 0.5p$	**1.467**	**1.251**	**1.098**	**1.121**
SGGM, $n = p$	**1.364**	**1.109**	**1.062**	**1.188**
SGGM, $n = 1.5p$	**1.362**	**1.198**	**1.098**	**1.025**
SGGM, $n = 2p$	**1.200**	**1.121**	**1.013**	**0.987**
CN	1.641	1.291	1.320	1.485
Salton	1.641	1.291	1.320	1.485
Jaccard	1.641	1.291	1.320	1.485
Sørensen	1.863	1.672	1.695	1.826
HPI	1.641	1.291	1.320	1.485
HDI	1.641	1.291	1.320	1.485
LHN	1.641	1.291	1.320	1.485
AA	1.641	1.291	1.320	1.485
RA	1.641	1.291	1.320	1.485
PA	2.512	1.251	1.340	1.446
LNBCN	1.641	1.291	1.261	1.382
LNBAA	1.641	1.291	1.261	1.382
LNBRA	1.641	1.291	1.261	1.382

Table 8: Basic topological features of four large networks.

Network	N	M	E	C	Avg_D	r
Email	1133	5451	0.299	0.220	9.622	0.078
Protein	1870	2240	0.099	0.099	2.396	−0.156
Grid	4941	6594	0.063	0.107	2.669	0.003
Astro-ph	18771	198050	0.091	0.486	4.140	0.294

Table 9: Comparison of 14 methods on 4 large datasets on AUC.

Method	Email	Protein	Grid	Astro-ph
SGGM	**0.916**	**0.943**	**0.944**	**0.929**
CN	0.817	0.566	0.570	0.864
Salton	0.812	0.566	0.569	0.863
Jaccard	0.475	0.479	0.487	0.479
Sørensen	0.547	0.493	0.529	0.584
HPI	0.809	0.566	0.569	0.863
HDI	0.813	0.566	0.569	0.863
LHN-I	0.805	0.565	0.570	0.864
AA	0.819	0.565	0.570	0.876
RA	0.818	0.566	0.568	0.838
PA	0.824	0.820	0.713	0.785

can efficiently recover the original network and predict the missing links accurately. As observed from Table 9, AUC of SGGM is highest on all 4 datasets. For Email, Protein, Grid, and Astro-ph datasets, SGGM improves 9.2%, 12.3%, 23.1%, and 5.3% on AUC, respectively. For error rate metric, as shown in Table 10, the SGGM method performs optimally with all 4 large datasets. It indicates that SGGM method could accurately recover the original network.

Table 10: Comparison of 14 methods on 4 large datasets on error rate.

Method	Email	Protein	Grid	Astro-ph
SGGM	**0.467**	**0.411**	**0.367**	**0.359**
CN	2.348	2.189	1.670	1.568
Salton	2.978	2.189	1.767	1.549
Jaccard	2.982	2.189	1.450	1.548
Sørensen	3.047	2.555	2.071	2.771
HPI	2.578	2.189	1.670	1.610
HDI	2.978	2.189	1.237	1.854
LHN-I	2.428	2.189	1.692	1.612
AA	2.922	2.204	1.872	1.953
RA	2.348	2.189	1.670	1.410
PA	9.219	8.032	15.081	12.451

4.6. Analysis. In the comparative analysis of the performance of the 14 methods using the eight real-world datasets, the SGGM method was outstanding in terms of AUC and error rate. This method can accurately recover the original adjacency matrix and, in most of cases, requires fewer samples, that is, far fewer than the actual number of nodes. The SGGM method's prediction precision gradually increased with the increase in number of samples; moreover, the recovery error rate decreased with the increase in matrix sparsity. Therefore, the SGGM method can be applied to small samples; however, it performs better with more samples. These results demonstrate that the SGGM method can be implemented in a scalable fashion.

5. Conclusions

Link prediction is a basic problem in complex network analysis. In real-world networks, obtaining node and edge properties is cumbersome. Link prediction methods based on network structure similarity do not depend on node properties; however, these methods are limited by network structure and are difficult to adapt in different network structures.

Our work is not dependent on node properties and expands link prediction methods that cannot deal with diverse network topologies. We sampled the original network adjacency matrix and used the SGGM method to depict the network structure.

Most nodes are independent in the actual network; thus, we used the SGGM method to estimate a precision matrix of the adjacency matrix to predict links. Our experimental results show that using the SGGM method to recover network structure is better than 13 mainstream similarity methods. We tested the proposed method with eight real-world datasets and used the AUC and error rate as evaluation indexes. We found that this method obtains higher prediction precision. We also analysed the influence of different parameters on the SGGM method and found no significant influence. The SGGM method returns a high AUC value within a certain range. Furthermore, the proposed method retains high prediction precision with fewer training samples; it can be applied in large network.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank Longqi Yang, Zhisong Pan, and Guyu Hu for their contributions to work on link prediction and Zhen Li and Junyang Qiu for their valuable discussions, comments, and suggestions. This work was supported by the National Technology Research and Development Program of China (863 Program) (Grant no. 2012AA01A510), the National Natural Science Foundation of China (Grant no. 61473149), and the Natural Science Foundation of Jiangsu Province, China (Grant no. BK20140073).

References

[1] L. L. Lü and T. Zhou, "Link prediction in complex networks: a survey," *Physica A: Statistical Mechanics & Its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[2] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining & Knowledge Discovery*, vol. 29, no. 1, pp. 203–236, 2013.

[3] Y. Yang, N. Chawla, Y. Sun, and J. Hani, "Predicting links in multi-relational and heterogeneous networks," in *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM '12)*, pp. 755–764, Brussels, Belgium, December 2012.

[4] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[5] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, "Evaluating link prediction methods," *Knowledge and Information Systems*, vol. 45, no. 3, pp. 751–782, 2015.

[6] N. Z. Gong, A. Talwalkar, L. MacKey et al., "Joint link prediction and attribute inference using a social-attribute network," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, article 27, 2014.

[7] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[8] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, pp. 25–102, 2001.

[9] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, NY, USA, 1986.

[10] P. Jaccard, "Etude comparative de la distribution florale dans une portion des Alpes et du Jura," *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37, no. 142, pp. 547–579, 1901.

[11] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biologiske Skrifter*, vol. 5, pp. 1–34, 1948.

[12] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Physical Review E*, vol. 73, no. 2, Article ID 026120, 2006.

[13] Y.-B. Xie, T. Zhou, and B.-H. Wang, "Scale-free networks without growth," *Physica A: Statistical Mechanics and Its Applications*, vol. 387, no. 7, pp. 1683–1688, 2008.

[14] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, pp. 623–630, 2009.

[15] Z. Liu, Q. M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks: a local naïve Bayes model," *EPL (Europhysics Letters)*, vol. 96, no. 4, Article ID 48007, 2011.

[16] J. Dahl, L. Vandenberghe, and V. Roychowdhury, "Covariance selection for nonchordal graphs via chordal embedding," *Optimization Methods & Software*, vol. 23, no. 4, pp. 501–520, 2008.

[17] O. Banerjee, L. E. Ghaoui, and A. Daspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Machine Learning Research*, vol. 9, no. 3, pp. 485–516, 2008.

[18] C. J. Hsieh, M. A. Sustik, I. S. Dhillon, P. K. Ravikumar, and R. Poldrack, "Big & quic: sparse inverse covariance estimation for a million variables," in *Advances in Neural Information Processing Systems*, pp. 3165–3173, MIT Press, 2013.

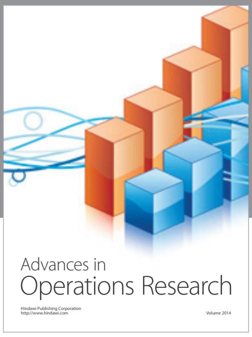
[19] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "QUIC: quadratic approximation for sparse inverse covariance estimation," *Journal of Machine Learning Research*, vol. 15, pp. 2911–2947, 2014.

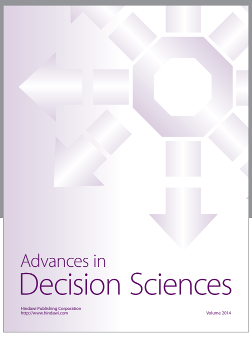
[20] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Physical Review Letters*, vol. 87, Article ID 198701, 2001.

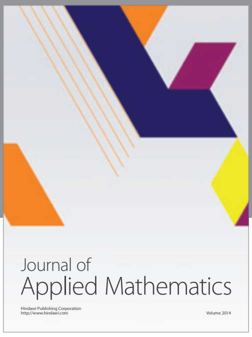
[21] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

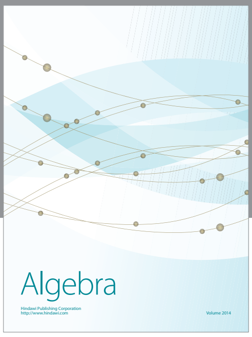
[22] M. E. J. Newman, "Assortative mixing in networks," *Physical Review Letters*, vol. 89, no. 20, Article ID 208701, 2002.

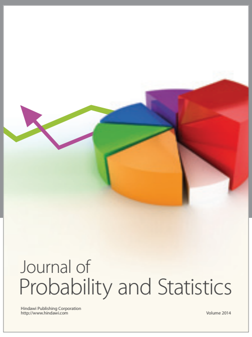
[23] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, vol. 6, Arizona State University, Tempe, Ariz, USA, 2009.


Advances in
Operations Research

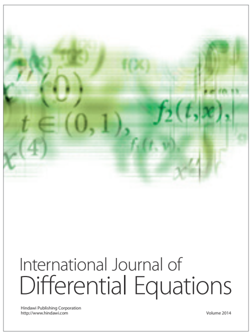

Advances in
Decision Sciences


Journal of
Applied Mathematics

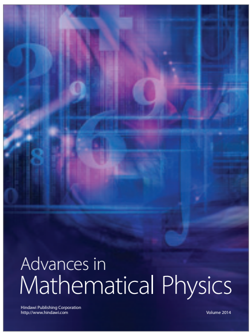

Algebra

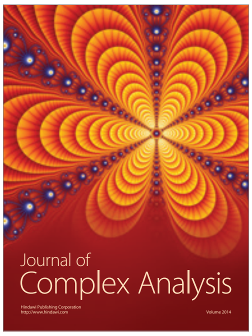

Journal of
Probability and Statistics

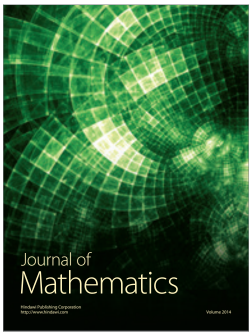
The Scientific
World Journal

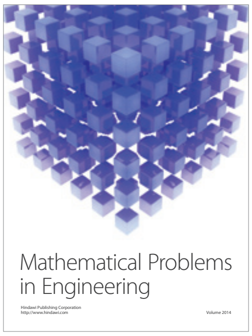

International Journal of
Differential Equations

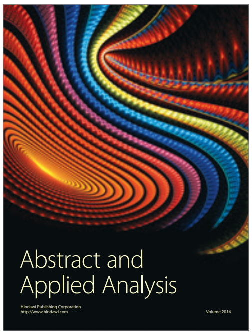
International Journal of
Combinatorics

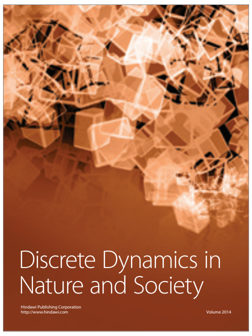

Advances in
Mathematical Physics

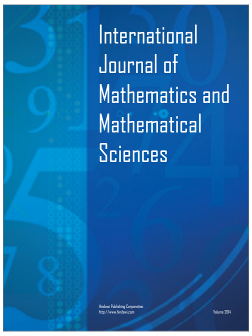

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

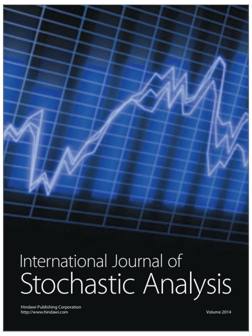

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization