IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingenierie des Systemes d'Information
Vol. 24, No. 1, February, 2019, pp. 35-42
Journal homepage: http://iieta.org/Journals/isi

Network Adjacency Matrix Blocked-compressive Sensing: A Novel Algorithm for Link Prediction

Fei Cai[1*], Xiaohui Mou[1], Xin Zhang[1], Jie Chen[1], Jin Li[1], Wenpeng Xu[2]

[1] College of Surveying and Geo-Informatics, Shandong Jianzhu University, Jinan 250101, China;
[2] Telchina Smart Industry Group Company Limited, Jinan 250101, China

Corresponding Author Email: caifei@sdjzu.edu.cn

https://doi.org/10.18280/isi.240104

Received: 8 December 2018
Accepted: 3 February 2019

Keywords:
compressive sensing (CS), measurement matrix, adjacency matrix, link prediction, subspace pursuit (SP)

ABSTRACT

Link prediction for complex networks is a research hotspot. The main purpose is to predict the unknown edge according to the structure of the existing network. However, the edges in real-world networks are often sparsely distributed, and the number of unobserved edges is usually far greater than that of observed ones. Considering the weak performance of traditional link prediction algorithms under the above situation, this paper puts forward a novel link prediction algorithm called network adjacency matrix blocked-compressive sensing (BCS). Firstly, the diagonal blocks were subjected to sparse transformation with the network adjacency matrix; Next, the measurement matrix was rearranged into a new measurement matrix using the sorting operator; Finally, the subspace pursuit (SP) algorithm was introduced to solve the proposed algorithm. Experiments on ten real networks show that the proposed method achieved higher accuracy and consumed less time than the baseline methods.

1. INTRODUCTION

Link prediction for complex networks has become a research hotspot in recent years. The purpose is to identify unobserved edges in the existing network or forecast future edges based on the current network structure [1]. Link prediction has been applied to recommend friends or interest points in social networks [2], predict the possible protein interactions without costly experiments in biology [3], simulate the drug interactions through a limited number of tests in medical pharmacy [4], and analyze the structure of criminal and terrorist networks to combat organized crime [5]. The existing methods for link prediction are either based on similarity or grounded on probability.

The similarity-based methods [2] assume that two nodes with a high similarity are more likely to be edged, and characterize the similarity between different nodes by local and global indices. The common indices include common neighbors (CN) index [6], adamic-adar (AA) index [7], local path (LP) index [8], and resource allocation (RA) index [9]. The methods using local indices enjoy fast speed and high efficiency. However, their prediction is not sufficiently accurate, as the node similarity is only constrained by local information. By contrast, the methods using global indices, e.g. Katz index [10], consumes lots of time to process large networks, despite its good accuracy.

The probability-based methods often assume that the network has a known structure. Under this assumption, the model is built and the parameters are estimated by statistical methods. The common probability-based models are hierarchical structure model [11] and random block model [12]. These methods boast many advantages in network analysis. However, the parameter learning and reasoning greatly increase the computing complexity, which limits the application range of such methods.

Both the similarity- and probability-based methods predict the possibility of unobserved edges between network nodes according to the structure of the existing network. Nevertheless, many real networks are so sparse that the edges of observed nodes only account for a small portion of network edges [13]. This poses a huge challenge to the existing link prediction methods, calling for the improving the link prediction in sparse network.

A possible way of improvement lies in the compressive sensing (CS) theory. Since it was proposed by Donoho [14], the CS has been implemented mainly in the processing of signals and images. The basic idea is that, in a proper low-dimensional representation, the information needed for a signal is fully contained in its under-sampled data. In other words, a signal can be reconstructed from a small set of sampled data, which is often the case in real networks with sparse edges. Considering the poor effect of existing link prediction methods in sparse network, this paper puts forward a novel link prediction algorithm called network adjacency matrix blocked-compressive sensing (BCS). Firstly, the diagonal blocks were subjected to sparse transformation with the network adjacency matrix; Next, the measurement matrix was rearranged into a new measurement matrix using the sorting operator; Finally, the subspace pursuit (SP) algorithm was introduced to solve the proposed algorithm. Experiments on ten real networks show that the proposed method achieved higher accuracy and consumed less time than the baseline methods.

2. PROBLEM STATEMENT AND PROPOSED METHOD

2.1 Introduction to the CS

In the CS theory [13], the measurement $x \in R^n$ can be defined as:

$$x_{m\times1} = A_{m\times n} b_{n\times1} \tag{1}$$

where $m << n$; A is the measurement matrix; $b \in R^n$ is the original signal. The sparsest solutions can be obtained by:

$$\min_b \|b\|_0 \quad s.t. \quad Ab = x \tag{2}$$

Since it is an NP-hard problem, (2) can be solved by l$_1$-minimization:

$$\min_b \|b\|_1 \quad s.t. \quad Ab = x \tag{3}$$

From (2) and (3), we have:

$$\min_b \|b\|_1 + \|Ab - x\|_2^2 \tag{4}$$

Here, (4) is known as the least absolute shrinkage and selection operator (LASSO). After introducing the sparseness control parameter λ, (4) can be rewritten as:

$$\hat{b} = \arg\min_b \lambda \|b\|_1 + \|Ab - x\|_2^2 \tag{5}$$

The construction of measurement matrix is shown in Figure 1 below.

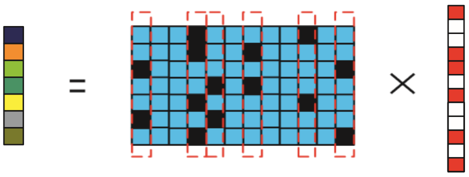


Figure 1. Construction of measurement matrix

2.2 Problem statement

In graph theory, an undirected network can be described as G=(V, E), where V and E are the set of nodes and set of edges, respectively. In the network, the number of nodes and the number of edges can be denoted as N=|V| and M=|E|, respectively. Let X be the adjacency matrix of the network. If there are edges between nodes i and j, then x$_{ij}$=x$_{ji}$=1; otherwise, x$_{ij}$=x$_{ji}$=0.

For each dataset, the network edges were divided into a training set E$_{train}$ and a test set E$_{test}$. Apparently, E$_{train}$ ∪ E$_{test}$=E and E$_{train}$ ∩ E$_{test}$ =∅. Let X$_{train}$ and X$_{test}$ be the adjacency matrix of the training set and the test set, respectively. Both matrices consist of 1 or 0 elements. Assuming that L=|E$_{test}$| is the number of edges in the test set, then the number of edges in the training set can be expressed as |E$_{train}$|=M-L. In addition, the number of all possible edges in the network is denoted as the candidate set $|\bar{E}|$=$\frac{N(N-1)}{2}$-(M-L). During the solution, the model was trained with the training set E$_{train}$, each possible edge between the nodes in the candidate set were given a score, and the scores were ranked in descending order. In this case, the node pair on the top has the largest probability of edge connection. Then, the test set E$_{test}$ results were verified by different evaluation metrics.

2.3 Proposed method

Let X be a $c \times r$ adjacency matrix of the network. First, the matrix was divided into n $B \times B$ matrix blocks, and the column vector of the i-th block was labelled x_i, where i=1, 2, 3, …, n, $n = c \times r/B^2$. Under the same conditions, x_i was measured by measurement matrix ϕ_B. Then, the vector of the measured value y_i was obtained, where the vector length is M_B ($M_B << B^2$). The vector y_i can be expressed as:

$$y_i = \phi_B x_i \quad (i = 1, 2, 3, ..., n) \tag{6}$$

where the $M_B \times B^2$ matrix ϕ_B is a Gaussian matrix. The original adjacency matrix x can obtain m CS values. For the original adjacency matrix, the total measurement matrix ϕ is the block diagonal matrix. The values of diagonal elements ϕ_B can be expressed as:

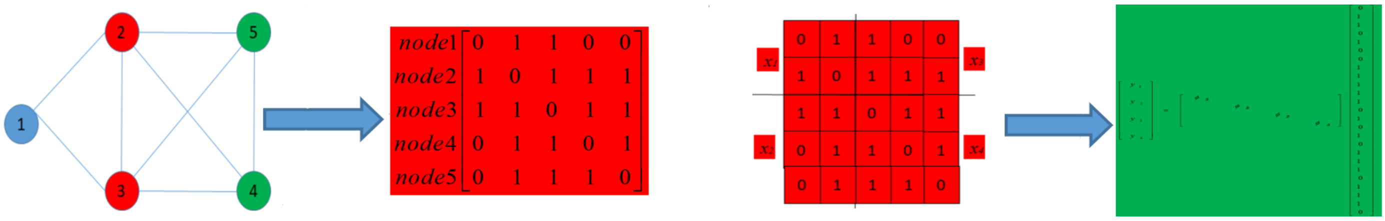
$$\phi = \begin{bmatrix} \phi_B & & & & & & \\ & \phi_B & & & & & \\ & & \phi_B & & & & \\ & & & \phi_B & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & & \phi_B \end{bmatrix} \tag{7}$$

The above analysis shows that the total measurement matrix ϕ can be stored with $M_B \times B^2$ matrix by our method, rather than with the $B \times B$ matrix. The required storage space is very small when B is small, which accelerate the implementation.

In our method, the matrix is solved by l$_1$-norm [13]:

$$\hat{b}_i = \arg\min_{b_i} \lambda \|b_i\|_1 + \|Ab_i - x_i\|_2^2$$
$$s.t.\ y_i = \phi_B x_i = \phi_B \psi \hat{b}_i \quad i = 1, 2, 3, ..., n \tag{8}$$

where ψ is the transformation matrix. To explain the principle of CS with matrix block, the network in Figure 2(a) was cited as an example. The construction of the measurement matrix is illustrated in Figure 2 below.



(a) Network (b)Adjacency matrix X (c) Block matrix (d) Measurement matrix

Figure 2. Construction of measurement matrix

The network adjacency matrix X in Figure 2 can be expressed as:

$$X = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} x_1 & x_3 \\ x_2 & x_4 \end{bmatrix} \tag{9}$$

where x_i are matrix blocks. According to Figure 2, the variable y can be described as:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \begin{bmatrix} \phi_B x_1 \\ \phi_B x_2 \\ \phi_B x_3 \\ \phi_B x_4 \end{bmatrix} = \begin{bmatrix} \phi_B & & & \\ & \phi_B & & \\ & & \phi_B & \\ & & & \phi_B \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \phi \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{10}$$

Sorting operator R, the vector $(x_1, x_2, x_3, x_4)^T$ can be rearranged into adjacent matrix:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = R \cdot X \tag{11}$$

The following can be derived from (10) and (11):

$$y = \phi R \cdot X = \phi \psi \cdot X \tag{12}$$

Thus, the similarity adjacency matrix $x^* \in R^N$ can be obtained [14]:

$$\begin{aligned} X^* &= \arg\min_{x_i} \lambda \| x_i \|_1 + \| \phi \psi x_i - y_i \|_2^2 \\ s.t. \quad y_i &= \phi x_i = \phi_B \psi x_i^* \quad i = 1, 2, 3, \ldots, n \end{aligned} \tag{13}$$

The l_1-norm minimization cannot guarantee the nonnegativity of the solution, but the size of network adjacency matrix must be nonnegative. To solve the problem, the SP algorithm [16, 17] was employed to recalibrate the prediction results.

Taking edges as the unknown vectors, a linear system was formed to find a solution to link prediction. The proposed method can identify unobserved edges in the existing network or forecast future edges based on the current network structure. The details on our method are given below.

Algorithm 1 BCS algorithm

```
Input: Undirected network G=(V, E)
Measurement matrix $\phi \in R^{M \times N}$
Vector $y \in R^M$
Network adjacency matrix $X \in R^N$
Sparsity K for network adjacency matrix
Output: Similarity adjacency matrix $X^* \in R^N$
Steps:
1. Initialize error $r_0 = y$, index set $\Lambda_0 = \phi$, and number of iterations n=1;
2. Generate intermediate candidate vectors $u_n = \phi^T r_{n-1}$;
3. Judge reliability: $\Lambda_n = sup((u_n)|_K)$, find the index with maximum K;
4. Update index: $\bar{\Lambda}_n = \Lambda_n \cup \Lambda_{n-1}$
5. Update intermediate candidate vectors: $\theta_n = \phi_{\bar{\Lambda}_n} y$;
6. Update candidate vectors: $\widehat{\theta}_n = (\theta_n)|_K$;
7. Update error: $r_n = y - \phi\theta_n$
8. Update the number of iteration n=n+1; If n<<K, go to step2; Otherwise, execute step 9;
9. Output similarity adjacency matrix $X^* \in R^N$
```

3. EVALUATION METRICS AND DATA RESOURCE

3.1 Evaluation metrics

Two evaluation metrics, namely, area under the curve (AUC) [18] and precision [19], were set up to compare the performance of our method with that of baseline methods. The two metrices are defined as follows.

(1) AUC

The AUC index measures the overall accuracy of the algorithm. It can be understood as the probability that a randomly selected missing edge has a higher score than a randomly chosen non-existent edge out of all unobserved edges. The AUC can be expressed as:

$$AUC = \frac{n' + n''}{n} \tag{14}$$

where n is the number of independent comparisons; n' is that the probability estimate of randomly selecting an edge in the n' test set is greater than that of randomly selecting an edge in the E test set; n'' is that the probability estimate of randomly selecting an edge in the n" test set equals that of randomly selecting an edge in the nonexistent edge set.

If all the scores are generated from an independent, identical distribution, the AUC value will approximate 0.5. Therefore, AUC>0.5 shows how much the algorithm outperforms random selection.

(2) Precision

Precision is defined as the ratio of the most likely predicted accuracy in the first L predicted edges given by the algorithm:

$$Precision = \frac{L_r}{L} \tag{15}$$

where L is the size of the predicted edges; L_r is the size of correctly predicted edges. Obviously, the higher the precision, the better the accuracy.

3.2 Baseline algorithms

The performance of our algorithm was compared with that of six typical similarity algorithms, i.e. the baseline algorithms, namely, common neighbors (CN) [6], resource allocation (RA) [9], adamic-adar (AA) [7], preferential attachment (PA) [20], local path (LP) [8] and Katz 10]. These similarity indices are described in details in Table 1.

Table 1. Six classical similarity indices

Name	Score	Description
CN	$S_{xy}^{CN} = \lvert\Gamma(x) \cap \Gamma(y)\rvert$	$\Gamma(x)$ and $\Gamma(y)$ are the sets of neighbors of node x and node y, respectively.
RA	$S_{xy}^{RA} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{k_z}$	k_z is the degree of node z.
AA	$S_{xy}^{AA} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\log k_z}$	k_z is the degree of node z.
PA	$S_{xy}^{PA} = k_x k_y$	k_x is degree of node x.
LP	$S_{xy}^{LP} = (A^2 + \alpha A^3)_{xy}$	α is the adjustable parameter,A is the adjacency matrix
Katz	$S_{xy}^{Katz} = ((I - \beta A^{-1}) - I)_{xy}$	β is the adjustable parameter, I is the diagonal matrix

3.3 Experiment data

Table 2. The statistics of the real-world networks

	|V|	|E|	C	r	<k>	H
FoodWeb	128	2075	0.335	-0.112	32.422	1.237
Karate	34	78	0.571	-0.476	4.588	1.693
Jazz	198	2742	0.617	0.020	27.697	1.395
USAir	332	2126	0.625	-0.208	12.807	3.464
Neural	297	2148	0.292	-0.163	14.465	1.801
Metabolic	453	2025	0.646	-0.226	8.940	4.485
Email	1133	5451	0.220	0.078	9.622	1.942
PB	1490	16715	0.263	-0.221	22.436	3.622
Yeast	2361	6646	0.130	-0.099	5.630	2.944
Router	5022	6258	0.012	-0.138	2.492	5.503

Note: |V| and |E| are the set of nodes and the set of edges, respectively; C is the clustering coefficient; r is the degree-degree correlation coefficient; <k> is the average degree. H is the degree heterogeneity.

To verify the performance of our algorithm, ten real networks from different domains were selected for experiments, including FoodWeb [21], Karate [22], Jazz [23], USAir [24], Neural [25], Metabolic [26], Email [27], PB [28], Yeast [29] and Router [30]. The statistics on the real-world networks are listed in Table 2 above.

4. RESULTS AND ANALYSIS

As mentioned before, our method was compared with six classical method through experiments on ten typical networks. Firstly, the observed edges were randomly divided into a training set and a test set. The training set was used to establish the prediction model, while the test set was adopted to verify the link prediction accuracy in complex networks. Next, the typical networks were transformed into undirected networks. Tables 3 and 4 compare the AUC and precision, two metrics of prediction accuracy, of the contrastive algorithms, respectively. Both the AUCs and precisions are the average values of 100 runs. The optimal AUC or precision on the ten networks are bolded, and the proportion of training set is 90%.

Table 3. The AUCs of different methods

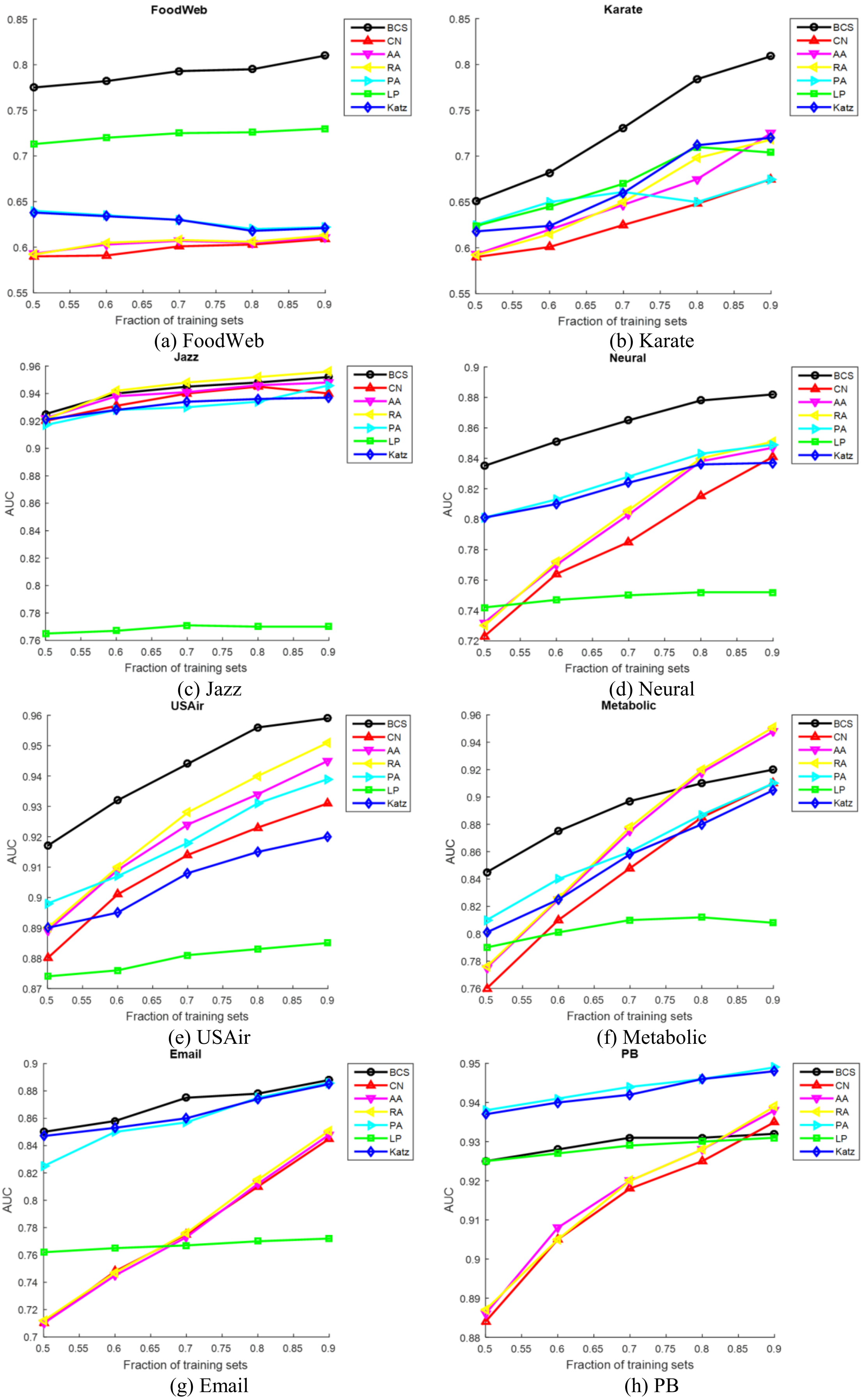
Network	BCS	CN	AA	RA	PA	LP	Katz
FoodWeb	**0.8152**	0.6104	0.6094	0.6120	0.7342	0.6223	0.6745
Karate	**0.8116**	0.6994	0.7341	0.7283	0.7008	0.7209	0.7371
Jazz	0.9663	0.9545	0.9612	**0.9701**	0.7668	0.9591	0.9485
USAir	**0.9599**	0.9355	0.9474	0.9537	0.8856	0.9427	0.9242
Neural	**0.8847**	0.8441	0.8589	0.8644	0.7529	0.8595	0.8575
Metabolic	0.9319	0.9198	0.9506	**0.9544**	0.8174	0.9243	0.9197
Email	0.8973	0.8442	0.8464	0.8467	0.7779	**0.8974**	0.8942
PB	0.9336	0.9361	0.9392	0.9393	0.9327	0.9495	**0.9500**
Yeast	**0.8412**	0.7061	0.7066	0.7061	0.7865	0.8357	0.8184
Router	**0.6654**	0.5580	0.5579	0.5579	0.4694	0.6320	0.3738

Table 4. The precisions of different methods

Network	BCS	CN	AA	RA	PA	LP	Katz
FoodWeb	**0.1762**	0.0707	0.0755	0.0754	0.1607	0.0758	0.1023
Karate	0.1487	0.1525	0.1538	0.1538	0.0863	**0.1750**	0.1613
Jazz	**0.6225**	0.5041	0.5242	0.5391	0.1304	0.5126	0.4920
USAir	0.3905	0.3730	0.3898	**0.4505**	0.3164	0.3738	0.3695
Neural	**0.1262**	0.0962	0.1039	0.1025	0.0575	0.0981	0.1027
Metabolic	**0.2113**	0.1378	0.1932	0.2680	0.0999	0.1449	0.1408
Email	**0.1503**	0.1392	0.1552	0.1401	0.0174	0.1467	0.1355
PB	0.0861	0.1729	0.1716	0.1493	0.0652	0.1735	**0.1744**
Yeast	**0.1070**	0.0924	0.0932	0.0741	0.0093	0.0950	0.0935
Router	**0.0253**	0.0168	0.0162	0.0098	0.0096	0.0214	0.0227

It can be seen that the proposed BCS outperformed the baseline methods under the AUC metric in 6 out of the 10 networks, namely, FoodWeb, Karate, USAir, Neural, Yeast and Router. Under the precision metric, our method outperformed the other methods in 7 out of the 10 networks, including FoodWeb, Jazz, Neural, Metabolic, Email, Yeast and Router. In particular, our method achieved the best values in large sparse networks like Yeast and Router.

The AUC and precision values of all the seven methods in the ten networks were further compared using different proportions of training set. The proportion of the training set was changed from 0.5 to 0.9. The results in Figures 3 and 4 show that our method achieved the best prediction accuracy and robustness among all methods.



(a) FoodWeb

(b) Karate

(c) Jazz

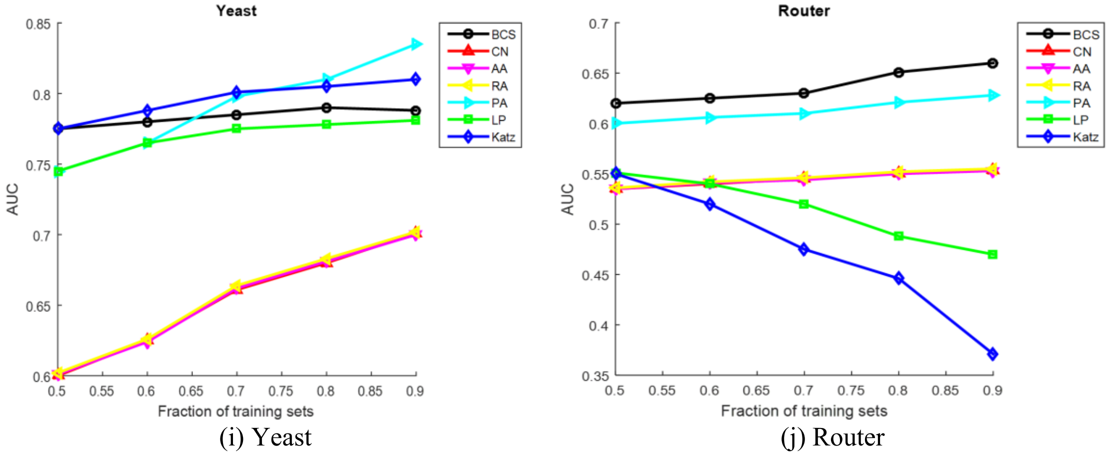
(d) Neural

(e) USAir

(f) Metabolic

(g) Email

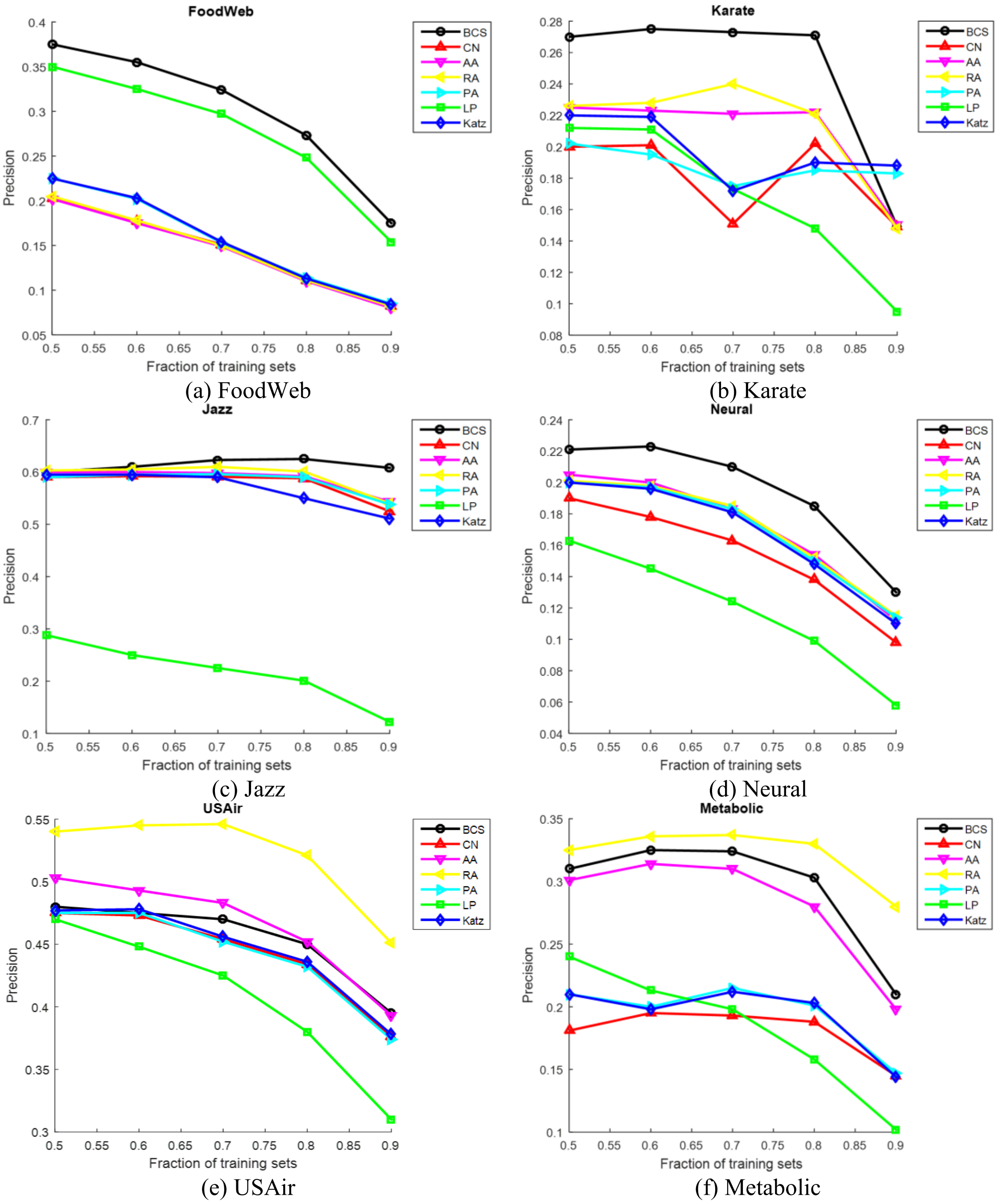
(h) PB



(i) Yeast

(j) Router

Figure 3. The AUCs of different methods under different proportions of training set



(a) FoodWeb

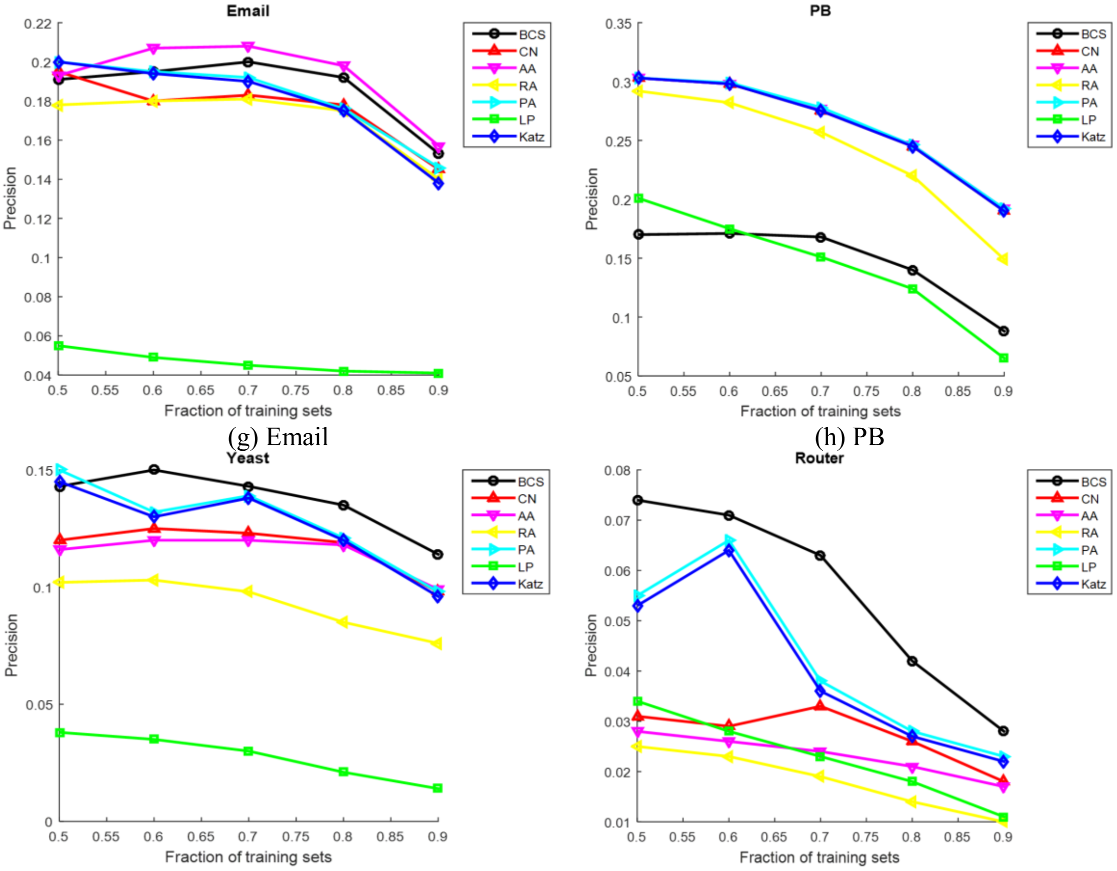
(b) Karate

(c) Jazz

(d) Neural

(e) USAir

(f) Metabolic



(g) Email

(h) PB

(i) Yeast

(j) Router

Figure 4. The precisions of different methods under different proportions of training set

Next, the computing time of our method was contrasted against that of the other methods through several experiment on a workstation with 64GB RAM and 3.10GHz 8-core processor. The comparison result is shown in Figure 5 below.

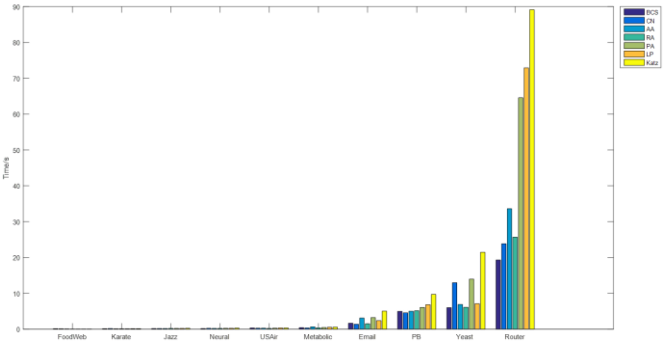


Figure 5. Comparison of computing time of different methods

As shown in Figure 5, the proposed method consumed much less time in computation than most of the contrastive methods. This proves the validity and reliability of our method.

5. CONCLUSIONS

Real-world networks are often very sparse, posing a severe challenge to traditional link prediction methods. What is worse, these methods usually have high complexity and low accuracy. To overcome these problems, this paper designs the BCS method for link prediction in complex networks, which uses the sorting operator to transform the measurement matrix into a new measurement matrix. The performance of the proposed method was verified by two evaluation metrics, the AUC and precision, through experiments on ten classical networks. The results show that our method outperformed the baseline methods in accuracy and computing time. Suffice it to say that this research designs a feasible, effective and competitive way to predict edges in complex networks.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (41601489), the Housing and Urban Rural Development Science and Technology Project of Shandong Province (2018-K2-03), University Science and technology project of Shandong Province (J18KB088), National College Students Innovation and Entrepreneurship Training project (201710430050).

REFERENCES

[1] Xie R, Shao ZF, Liu B, Xu SB. (2018). Application of asymmetric information in link prediction. Journal of Computer Applications 38(6): 1698-1702, 1708. https://doi.org/10.11772/j.issn.1001-9081.2017102467

[2] Shahmohammadi A, Khadangi E, Bagheri A. (2016). Presenting new collaborative link prediction methods for activity recommendation in Facebook. Neurocomputing, 210: 217-226. https://doi.org/10.1016/j.neucom.2016.06.024

[3] Li Y, Zhang XF, Yi M, Liu YY. (2015). Link prediction for the gene regulatory network. Acta Mathematica Scientia 35(5): 1018-1024. https://doi.org/10.3969/j.issn.1003-3998.2015.05.017

[4] Fokoue A, Hassanzadeh O, Sadoghi M, Zhang P. (2016). Predicting drug-drug interactions through similarity-based link prediction over web data. In International Conference Companion on World Wide Web, 2016: 175-178. https://doi.org/10.1007/978-3-319-34129-3_47

[5] Fan C, Liu Z, Lu X, Xiu BX, Chen Q. (2017). An efficient link prediction index for complex military organization. Physica A: Statistical Mechanics and its Applications 469: 572-587. https://doi.org/10.1016/j.physa.2016.11.097

[6] Lorrain F, White HC. (1971). Structural equivalence of individuals in social networks. The Journal of Mathematical Sociology 1(1): 49-80. https://doi.org/10.1080/0022250X.1971.9989788

[7] Adamic LA, ADAR E. (2003). Friends and neighbors on the Web. Social Networks 25(3): 211-230. 10.1016/S0378-8733(03)00009-1

[8] Lv L, Jin CH, Zhou T. (2009). Similarity index based on local paths for link prediction of complex networks. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics 80(4): 046122. https://doi.org/10.1103/physreve.80.046122

[9] Zhou T, Lv L, Zhang YC. (2009). Predicting missing links via local information. European Physical Journal B, 71(4): 623-630. https://doi.org/10.1140/epjb/e2009-00335-8

[10] Katz L. (1953). A new status index derived from sociometric analysis. Psychometrika 18(1): 39-43. https://doi.org/10.1007/BF02289026

[11] Clauset A, Moore C, Newman ME. (2008). Hierarchical structure and the prediction of missing links in networks. Nature 453(7191): 98-101. https://doi.org/10.1038/nature06830

[12] Guimerà R, Sales-Pardo M. (2009). Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences 106(52): 22073-22078. https://doi.org/10.1073/pnas.0908366106

[13] Costa M, Desmarais BA, Hird JA. (2016). Science use in regulatory impact analysis: The effects of political attention and controversy. Review of Policy Research 33(3): 251-269. https://doi.org/10.1111/ropr.12171

[14] Dnonho DL. (2006). Compressed sensing. IEEE Transactions on Information Theory 52(4): 1289-1306. https://doi.org/10.1109/TIT.2006.871582

[15] Meng J, Li H, Han Z. (2009). Sparse event detection in wireless sensor networks using compressive sensing. 2009 43rd Annual Conference on Information Sciences and Systems, pp. 181-185. https://doi.org/10.1109/CISS.2009.5054713

[16] Lee H, Lee DS, Kang H. (2011). Sparse brain network recovery under compressed sensing. IEEE Transactions on Medical Imaging 30(5): 1154-1165. https://doi.org/10.1109/TMI.2011.2140380

[17] Liben-Nowell D, Kleinbergl J. (2003). The link prediction problem for social networks. Journal of the American Society for Information Science and Technology 58(7): 556-559. https://doi.org/10.1002/asi.20591

[18] Hanley JA, Mcneil BJ. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143(1): 29-36. https://doi.org/10.1148/radiology.143.1.7063747

[19] Herlocker JL, Konstan JA, Terveen LG, Riedl, JT. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1): 5-53. https://doi.org/10.1145/963770.963772

[20] Barabasi AL, Albert R. (1999). Emergence of scaling in random networks. Science 286(5439): 509-509. https://doi.org/10.1126/science.286.5439.509

[21] Ulanowiza RE, Bondavalli C, EGNOTOVICH MS. (1998). Network analysis of trophic dynamics in South Florida ecosystem, FY 97: The Florida Bay Ecosystem.

[22] Zachary WW. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33(4): 452-473. https://doi.org/10.1086/jar.33.4.3629752

[23] Gleiser PM, Danon L. (2003). Community structure in jazz. Advances in Complex Systems 6(4): 565-573. https://doi.org/10.1142/S0219525903001067

[24] Batagelj V, Mrvar A. (2006). Pajek datasets.

[25] White JG, Southgate E, Thomson JN, Brenner S. (1986) The structure of the nervous system of the nematode Caenorhabditis elegans. Philosophical Transactions of the Royal Society of London B: Biological Sciences 1(3): 314-340. https://doi.org/10.1098/rstb.1986.0056

[26] Duch J, Arenas A. (2005). Community detection in complex networks using extremal optimization. Physical Review E Statistical Nonlinear and Soft Matter Physics 72(2): 027104. https://doi.org/10.1103/PhysRevE.72.027104

[27] Guimerà R, Danon L, Díaz-Guilera A, Giralt F, Arenas, A. (2003). Self-similar community structure in a network of human interactions. Physical Review E 68(6): 065103. https://doi.org/10.1103/PhysRevE.68.065103

[28] Adamic LA, Glance N. (2005). The political blogosphere and the 2004 U.S. election: Divided they blog. In Proceedings of the 3rd International Workshop on Link Discovery, New York, NY, USA, pp. 36-43. https://doi.org/10.1145/1134271.1134277

[29] von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, Bork P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417(6887): 399-403. https://doi.org/10.1038/nature750

[30] Spring N, Mahajan R, Wetherall D. (2002). Measuring ISP topologies with rocket fuel. In Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, New York, NY, USA, pp. 133-145.