

SCIENTIFIC REPORTS



Correction: Author Correction

OPEN

A Robust Method for Inferring Network Structures

Yang Yang^{1,2}, Tingjin Luo^{3,4}, Zhoujun Li¹, Xiaoming Zhang¹ & Philip S. Yu²

Inferring the network structure from limited observable data is significant in molecular biology, communication and many other areas. It is challenging, primarily because the observable data are sparse, finite and noisy. The development of machine learning and network structure study provides a great chance to solve the problem. In this paper, we propose an iterative smoothing algorithm with structure sparsity (ISSS) method. The elastic penalty in the model is introduced for the sparse solution, identifying group features and avoiding over-fitting, and the total variation (TV) penalty in the model can effectively utilize the structure information to identify the neighborhood of the vertices. Due to the non-smoothness of the elastic and structural TV penalties, an efficient algorithm with the Nesterov's smoothing optimization technique is proposed to solve the non-smooth problem. The experimental results on both synthetic and real-world networks show that the proposed model is robust against insufficient data and high noise. In addition, we investigate many factors that play important roles in identifying the performance of ISSS.

Almost everything in our daily life can be modeled as complex networks, such as the social network^{1,2}, transportation network³, protein-to-protein network⁴ and knowledge graph⁵. The interactions in these networks play an important role in identifying the networks' structure, functionality and dynamics. Hence, many researchers utilize the detailed interaction information to do a lot of interesting research works, such as the centrality measures^{6,7}, community detection⁸, robustness analysis^{9–11}, controllability¹² and network game^{13–15}. However, the link information of networks is often invisible in many cases. Hence, it's very important to propose a robust method in inferring the network structure from very few observed values (measurements) with noise, which are indirectly generated from the network.

Many researchers have made great efforts in solving the network inference problem. For instance, Han and Di *et al.*¹⁶ proposed the state-of-art method i.e., Lasso (Least Absolute Shrinkage and Selection Operator)¹⁷, which provides an estimation of a network with limited connectivity and low model prediction error. Hempte¹⁸ proposed an inference algorithm based on inner composition alignment to identify the network structure^{19,20} on the time series data²¹. Timme²² and Domenico Napoletani *et al.*²³ inferred the complete connectivity of a network from its stable response dynamics. Daniel Marbach *et al.*²⁴ summarized many robust gene network inference methods and compared their performances. Soheil Feizi *et al.*²⁵ proposed network deconvolution as a general method to distinguish direct dependencies in gene expression regulatory networks.

Though many researchers have made great efforts in solving the network inference problem, it is still very challenging mainly on the following reasons. Firstly, the interactions in the network are very sparse. For most of the vertices of the network, they have few interactions with their neighbors, which are only a small portion of vertices compared with the network size, i.e. the total number of vertices. It leads to the sparseness of interactions for each vertex, which makes it hard to infer the local structure. Secondly, the inference of the inaccessible original network is mainly based on the observations, i.e. measurements generated from the network. However, almost all of measurements and outputs have noises. The noisy data present great challenges in the error control, and it creates great difficulties to identify the real network. Thirdly, the observations are always insufficient i.e., have missing terms. In many real world applications, it's hard or costly to get all the measurements of the network from the beginning to the end. For many real world cases, we have only 5–20% observations¹⁶ on the whole network, which makes it an impossible task to infer the detailed network structure.

¹School of Computer Science and Engineering, Beihang University, Beijing, 100191, China. ²Department of Computer Science University of Illinois at Chicago, Chicago, 60607, United States. ³College of Science, National University of Defense Technology, Changsha, Hunan, 410073, China. ⁴Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, 48109, USA. Correspondence and requests for materials should be addressed to Z.L. (email: lizj@buaa.edu.cn)

In this paper, we propose an iterative smoothing algorithm with structure sparsity (ISSS) method to infer the network structure. The elastic penalty is the combination of L_1 and L_2 penalties. The special L_1 penalty can successfully model the sparseness of network information. The L_2 penalty can prevent over-fitting of the model. The TV penalty²⁶ of the method can treat the neighbors of each vertex as a continuous signal, which makes the model tend to infer the local information of the network, i.e. community structure. The three structural penalties lead to the non-smoothness of the loss function, which makes it hard to solve the optimization problem. By introducing the Nesterov's smoothing technique²⁷, the optimization problem can be successfully solved. Moreover, the existence of three penalties and the smoothing technique makes the inference method more robust, even when the network has an insufficient observation with noise. We apply the model on synthetic networks to infer the dynamics of these networks, such as Erdős Rényi (ER)²⁸, Barabási Albert (BA)²⁹ and Watts Strogatz (WS)³⁰ networks. Furthermore, we also apply the model on many real world networks, such as the Karate³¹ and Netscience³² networks. The experimental results show the effectiveness of the ISSS model.

Problem Definition

Let's take the ultimatum game (UG) on networks³³ as an example. The vertices in the network represent players. If two people play the ultimatum game, there is an edge between them. Everyone in the game will play two times with his neighbors using strategy (p, q) , both as a proposer and a responder. $p_{ij} \in [0, 1]$ is the amount proposed by player i to offer his neighbor j , while q_{ij} is the minimum amount that i responds to his neighbor j . For instance, player i has a strategy $[0.8, 0.65]$, and one of his neighbors j has a strategy $[0.6, 0.5]$. Since i proposes to offer $p_{ij} = 0.8$ to j , which is larger than the minimum acceptance amount of j , i.e., $q_{ji} = 0.5$, then player i will get a $1 - 0.8 = 0.2$ payoff. At the same time, j offers i with $p_{ji} = 0.6$, which is lower than the minimum amount of i accepts, i.e., $q_{ij} = 0.65$, then player i rejects j 's offer, and both of them get nothing. The game lasts for M rounds. For round $\mu \in [1, M]$ with time stamp t_μ , the corresponding strategy of player i is denoted as $(p_j(t_\mu), q_j(t_\mu))$. The payoff of player i with player j at round μ can be summarized as follows:

$$\phi_{ij}(t_\mu) = \begin{cases} p_j(t_\mu) + 1 - p_i(t_\mu) & p_i(t_\mu) \geq q_j(t_\mu) \text{ and } p_j(t_\mu) \geq q_i(t_\mu) \\ 1 - p_i(t_\mu) & p_i(t_\mu) \geq q_j(t_\mu) \text{ and } p_j(t_\mu) < q_i(t_\mu) \\ p_j(t_\mu) & p_i(t_\mu) < q_j(t_\mu) \text{ and } p_j(t_\mu) \geq q_i(t_\mu) \\ 0 & p_i(t_\mu) < q_j(t_\mu) \text{ and } p_j(t_\mu) < q_i(t_\mu) \end{cases} \quad (1)$$

Hence, the overall payoff of player i with all its neighbors is defined as $y_i(t_\mu) = \sum_{j \in E_i} \phi_{ij}(t_\mu)$, where E_i is the neighbors of player i . The strategy $(p_i(t_{\mu+1}), q_i(t_{\mu+1}))$ changes from time to time with a random number $\delta \in [-\varepsilon, \varepsilon]$. ε is set to 0.05 and the strategy p and q are set to range $[0, 1]$. The reconstruction problem is formulated as $Y_i = \Phi_i \times X_i$ as shown in Fig. 1, where $Y_i \in \mathbb{R}^{M \times 1}$ is given by the output of the network system for vertex i . $\Phi_i \in \mathbb{R}^{M \times N}$ is the observation, i.e. measurements which is constructed by Equation (1). $X_i \in \mathbb{R}^{N \times 1}$ denotes the neighboring vector of vertex i , which is the network structure we want to infer. $t_\mu \in [1, M]$ is the accessible time instance in time series. Our task is to infer all the neighbors of vertex i , i.e., X_i , given noisy, sparse and incomplete Y_i and Φ_i in Equation (2).

$$\begin{bmatrix} y_i(t_1) \\ y_i(t_2) \\ \vdots \\ y_i(t_M) \end{bmatrix} = \begin{bmatrix} \phi_{i1}(t_1) & \phi_{i2}(t_1) & \dots & \phi_{iN}(t_1) \\ \phi_{i1}(t_2) & \phi_{i2}(t_2) & \dots & \phi_{iN}(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{i1}(t_M) & \phi_{i2}(t_M) & \dots & \phi_{iN}(t_M) \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iN} \end{bmatrix} \quad (2)$$

Here the UG on networks is just an example. There are also many other applications, such as the evolutionary games, transportation, communication processes, sociology and biology in papers^{16,34,35}. For instance, the construction of biological interaction networks with the goal of uncovering causal relationships between genotype and phenotype constitutes a major research topic in systems biology³⁵. With the increased availability of DNA microarray time-series data, it is possible to infer gene regulatory networks (GRNs) from gene expression data to understand dynamic GRNs³⁶. The information-geometric network inference to use a real radio emulation testbed to infer the end-to-end rate distributions of stochastic network flows from link rate measurements³⁷. G Zahoranszky-Kohalmi³⁸ employs network inference method in investigating the drug-target and target-target interactions in order to design new drugs.

In these real world applications, the observations from the network, i.e., Φ_i and Y_i have noises, and Φ_i is generally sparse and incomplete, which are key factors in identifying the performance of the network inference method. In fact, the essence of the inferring network structure (network reconstruction) problem can be formulated by Equation (2). The performance of the model is determined by the network structure and the expressing ability of model itself. Thus if the networks are built using ER network model with the same parameters, then the performance of any methods should be very similar. In the next section, we will discuss the advantages of ISSS and the key factors in identifying the performance of the ISSS and Lasso models.

Results

In this section, we apply our model on many different types of synthetic and real world networks. Then we compare the experimental results of our model with the state-of-art method Lasso¹⁶ on ER, BA and WS networks. The experimental results show the effectiveness of our model, and it outperforms the baseline method significantly. In addition, we also investigate the key factors in identifying the performance of the model, such as the variance

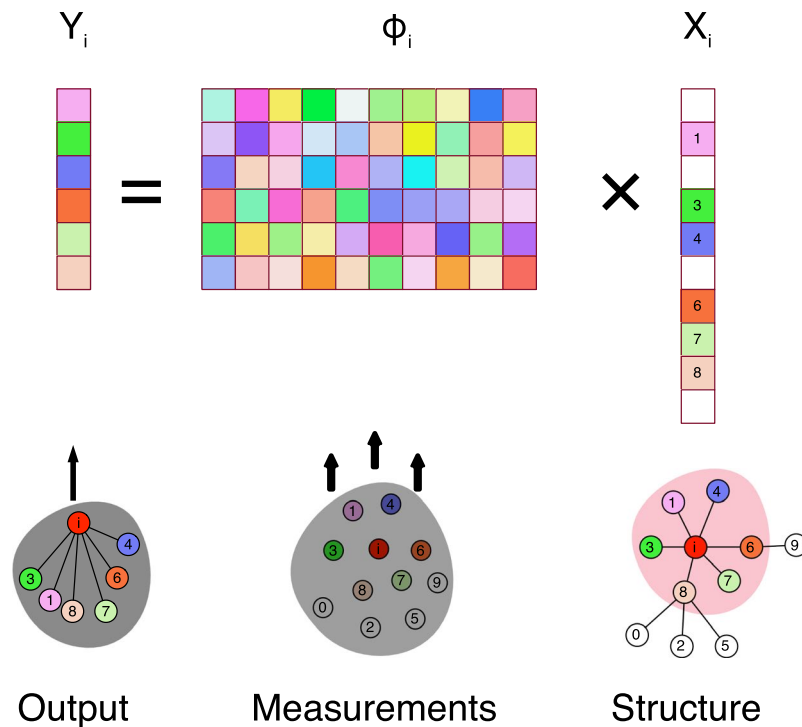


Figure 1. The network structure inference problem. The colorful squares in Y_i and Φ_i represent different float values. The colorful squares in X_i represent the connection between vertex i and other vertices. If the square of vertex j is blank, it means that there is no connection between vertex i and j . Hence, the color of squares in X_i is the same with the corresponding color of nodes in the graph. In many real world applications, the training data is insufficient, which indicates that some rows of Y_i and Φ_i are missing, such as the $j \in E_{missing}$ row in Y_i and Φ_i .

Type	Method	Data				
		0.05	0.1	0.15	0.2	0.4
ER	Lasso	0.51	0.55	0.59	0.64	0.95
	ISSS	0.63	0.71	0.76	0.80	0.95
BA	Lasso	0.53	0.63	0.81	0.92	0.99
	ISSS	0.73	0.79	0.86	0.92	1.00
WS	Lasso	0.53	0.60	0.71	0.87	1.00
	ISSS	0.71	0.79	0.86	0.93	1.00

Table 1. Comparison between ISSS and Lasso. We use AUROC³⁹ (Area under the Receiver Operating Characteristic) index to depict the performance of the inference task. Area is the area under the curve. It is a common summary statistic for the goodness of a predictor in a binary classification task. It is equal to the probability that a predictor will rank a randomly chosen positive instance higher than a randomly chosen negative one.

of the elements in measurement Φ and the properties of many real world networks. Finally, we compare the clustering coefficients of the network derived from the Lasso and ISSS model, respectively. And we also analyze the effectiveness of three penalties in the model.

Experiments on three typical networks. We take the network inference as a binary classification task, i.e. the edge exists or not. Hence, Area under the Receiver Operating Characteristic (AUROC)³⁹ is employed to evaluate the performance of the models. The Area under the Receiver Operating Characteristic is a common summary statistic for the goodness of a predictor in a binary classification task. It is equal to the probability that a predictor will rank a randomly chosen positive instance higher than a randomly chosen negative one. According to the experimental results in Table 1, the ISSS method outperforms Lasso on all the ER, BA and WS networks with different training data size, which is defined as $Data = M/N$, where M is the number of accessible time instances in the time series and the N is the number of vertices in the network. In the experiment, the ER, BA and WS networks all have $N = 100$ vertices, $M = 100$ time instances and the edges of networks are constructed randomly according to the parameters. All the experimental results are averaged over at least 10 independent trials. The experiment setup in this paper is the same with the state-of-art Lasso. With more than 40% of the training

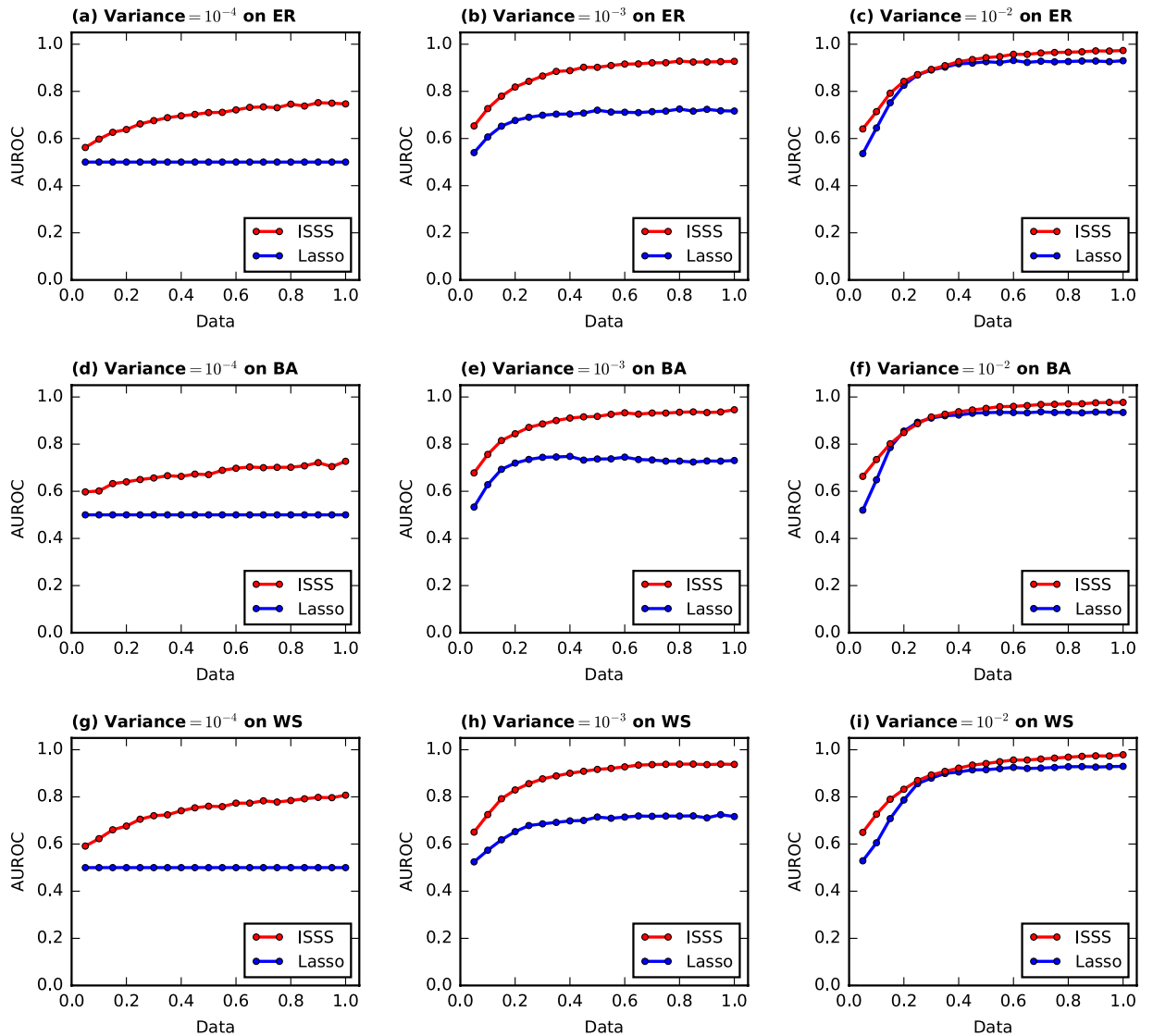


Figure 2. The performance of ISSS and Lasso on the data with $\text{variance} = 10^{-2}$, 10^{-3} and 10^{-4} . With very small $\text{variance} = 10^{-4}$, the performance of Lasso is equal to random guess, i.e., the blue line in Fig. 2(a,d,g). With very large variance, the inference or reconstruction task is much easier. To show the whole curves, the upper bound of the y axis is set to 1.2 for all the figures in this paper.

data, i.e. $\text{Data} = 0.4$, both ISSS and Lasso can successfully identify the network structure. As shown in Table 1, when the training data size is very small, the performance of ISSS is much better than Lasso. Take $\text{Data} = 0.05$ as an example, the AUROC of ISSS is at least 0.1 larger than that of Lasso. It indicates that the ISSS improves the performance significantly with very few training data.

It is worth mentioning that variance of measurement Φ 's elements in Equation (2) play an important role in inferring the network structure. The Φ 's elements $\Phi_{ij} \in [0, 1]$ on ER, BA and WS in Table 1 has default $\text{variance} = 10^{-1}$. Here we will analyze the situation that Φ has many other variances, s.t. $\text{variance} = 10^{-2}$, 10^{-3} and 10^{-4} in Fig. 2. We plot the performance of ISSS and Lasso with different variances of Φ 's elements on ER, BA and WS networks.

On the networks with $\text{variance} = 10^{-4}$ in Fig. 2(a,d,g), i.e. the network structure is hard to reconstruct, Lasso fails to infer the network structure. The performance of Lasso is the same with the random guess. Compared with the Lasso, ISSS method greatly outperforms the lasso significantly. As the size of training data goes up, the performance of ISSS gets better. However, the AUROC of ISSS is still lower than 0.8 with all the training data, i.e., $\text{Data} = 1$. On the networks with $\text{variance} = 10^{-3}$, as shown in Fig. 2(b,e,h), there is a big gap between two curves, which indicates that ISSS greatly outperforms the Lasso. With $\text{variance} = 10^{-2}$ as shown in Fig. 2(c,f,i), the performance of ISSS is greatly improved especially when we have only 5% training data. As the size of training data rises, both the performances of ISSS and Lasso are improved. An interesting phenomenon is that ISSS and Lasso perform similarly among 25–40% training data with $\text{variance} = 10^{-2}$. The cause of the phenomena is hard to interpret from theory and experiment aspects. In addition, low variance leads to faster convergence of the

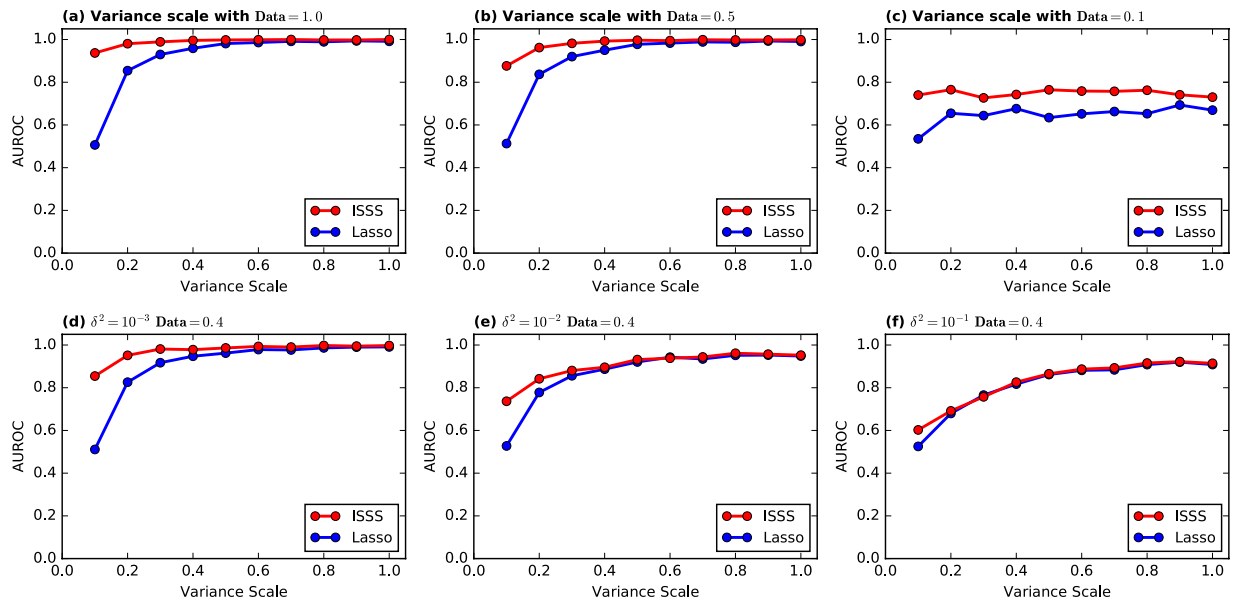


Figure 3. Insufficient data vs variance scale. In most cases, the AUROC of ISSS and Lasso rises, as the variance increases. With small variance, ISSS performs much better than the Lasso. With $Data = 0.1$, the performance cannot be improved with any variance. It indicates that insufficient data plays an important role in the inference task. Extremely insufficient data leads to the failure of all the methods. The experimental results with other $Data \in [0.2, 1]$ values are similar. In this section, we take $Data = 0.4$ as a representative. As shown in this figure, with small noise, ISSS outperforms Lasso. With great noise, the performance of ISSS and Lasso reach the upper bound of the model. Hence the performance of the two methods are the same.

Network	Vertices	Eedgs	CC	Average path	Centralization	Heterogeneity	Density	Lasso	ISSS
Karate	34	78	0.571	2.408	0.4	0.833	0.139	N/A	0.2
Dolphins	62	159	0.259	3.357	0.116	0.572	0.084	N/A	0.5
Netscience	1589	2743	0.638	5.823	0.019	1.005	0.002	N/A	0.8
Adjnoun	113	435	0.173	2.536	0.38	0.903	0.068	N/A	0.6
Football	115	613	0.403	2.508	0.012	0.083	0.094	N/A	1
Les Miserables	77	254	0.573	2.641	0.397	0.91	0.087	N/A	0.4

Table 2. Minimum data for achieving at least 0.95 AUROC in combination with several real networks. CC stands for clustering coefficient.

model. For instance, to infer a small network, the training time with $variance = 10^{-3}$ is 10% shorter than that with $variance = 10^{-1}$. That's the reason that many systems prefer low variance measurements and outputs. In summary, the ISSS model performs well with low variance measurements and few training data.

To further investigate the relation between variance and training data size, as shown in Fig. 3(a,b,c), we fix the training data size, i.e., $Data = 0.1, 0.5$ and 1, then find out the performance of ISSS and Lasso as variance scale goes up from (0, 1] with step 0.1. The experimental results on ER, BA and WS network are similar. Hence, we take ER network as an example in the figure. When $Data = 1$ and 0.5, ISSS can successfully infer the network structure with very small variance scale. With $Data = 0.1$, the AUROC keeps its value as the variance goes up. The performance of Lasso is just a little better than the random guess. However, the AUROC value of ISSS is larger than 0.70. It indicates that training data size is an important factor in identifying the upper bound of the performance.

To test the robustness of the ISSS against the noise, we add some Gaussian noises $\mathcal{N} \sim (\mu, \delta^2)$ to the output Y and measurement Φ . Then we apply the ISSS and Lasso model on the noisy data. The results are shown in Fig. 3(d,e,f). For low noise with $\delta^2 = 10^{-3}$, the ISSS outperforms the Lasso, especially when the elements of Φ have very small variance. For very high noise that follows the normal distribution with $\delta^2 = 10^{-1}$, the performance of ISSS and Lasso are similar. For the noise with $\delta^2 = 10^{-2}$, the performance of ISSS is better than Lasso for all variance scales.

Experiments on real world networks. We apply the ISSS and Lasso on many real world networks to evaluate the performance. Table 2 denotes the minimum data for achieving at least 0.95 AUROC in combination with several real world networks. Lasso cannot achieve 0.95 AUROC on all networks with default $variance = 10^{-1}$. ISSS greatly outperforms the Lasso. On karate and Les Miserables networks, ISSS achieves 0.95 AUROC with only $Data = 0.2$ and 0.4, respectively. By analyzing many properties of the network, we find that it is much easier to

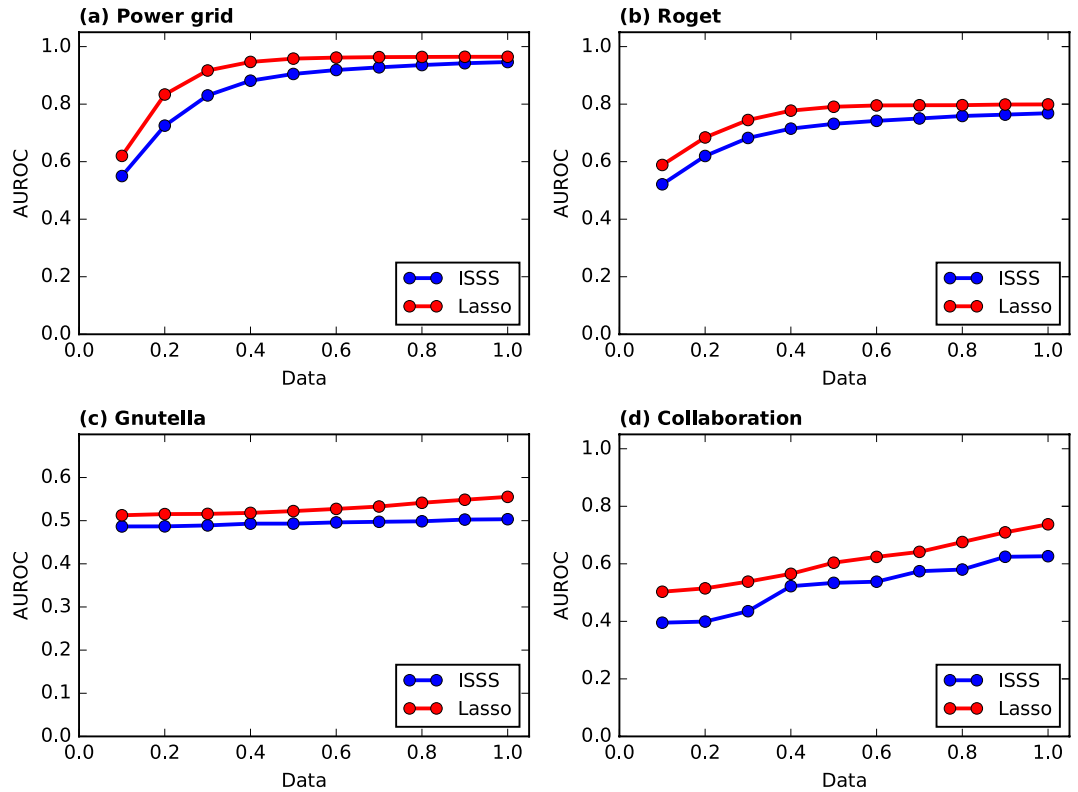


Figure 4. The performance of ISSS and Lasso on power grid, roget, Gnutella peer-to-peer file sharing network and collaboration network in computational geometry. The performance of ISSS and Lasso ON Gnutella network are not good enough. The reason is that we need more training data to infer the network, i.e., $Data = M/N = 2$, where M is the time instances and N is the number of vertices in the network.

infer the network structure with higher centralization index with very few training data. The network with both higher heterogeneity and density can also be easily inferred. To evaluate the performance of ISSS and Lasso on large networks, we do some experiments on the power grid (4941 vertices, 6594 edges), roget (1022 vertices, 5075 edges), Gnutella peer-to-peer file sharing network (6301 vertices, 20777 edges) and collaboration network in computational geometry (7343 vertices, 11898 edges). As shown in Fig. 4, ISSS outperforms the Lasso on all the networks. The Gnutella peer-to-peer network is hard to infer. Both ISSS and Lasso require more training data to infer the Gnutella peer-to-peer network structure.

We compare the clustering coefficient of the network inferred by ISSS and Lasso with different training data sizes in this part. In Fig. 5, the horizontal axis represents the training data size, and the vertical axis is the clustering coefficient of the network inferred by ISSS and Lasso. The red solid line lies above the blue line all the time. It indicates that the ISSS model is inclined to infer the networks with stronger community structures than that of Lasso. As the training data size increases in Fig. 5, the clustering coefficient of the network inferred by ISSS approaches the ground truth, i.e., the yellow line. These results show that the TV penalty in ISSS model works well. Here is an example to demonstrate why the TV penalty works. For instance, a network has 50 vertices. As shown in Fig. 6, the horizontal axis represents all the 50 vertices in the network. If vertex j is a neighbor of vertex i , it is 1, otherwise it is 0 in the figure. Lasso tends to predict the neighbors of vertex i discretely as shown in Fig. 6(a), while the ISSS is opt to infer the network structure in a continuous way, as shown in Fig. 6(b). The reason for choosing TV penalty is that almost all of the vertices in many networks have similar natural number. Take the karate network as an example. Almost all the vertices' labels that are less than 20 are in the same community. For the dolphin network, the vertices' labels under 30 are connected with each other. For the networks don't have natural number labels or the vertices don't have continuous labels as shown in Fig. 6(a). We can remove the TV penalty from the model. The experimental results are shown in Table 3. The experiment is done on a BA network with continuous natural number labels. Without TV penalty, the ISSS model also works well. However, the ISSS with all the three penalties is still the best.

As we mentioned above, TV penalty is a factor in inferring the network structure. Hence, how to choose the value of parameter γ is important as shown in Equation (4). In this part, we investigate the relationship between the most common used hyper parameter coefficients γ and the training data size, i.e. $Data$. According to the results in Table 4, $\gamma = 10^{-3}$ and 10^{-4} have the best performance on almost all training data. Given $Data = 0.1$, $\gamma = 10^{-6}$ is the best choice. The hyper parameter β , λ of L_1, L_2 are empirically set as 0.001 and 0.0001, respectively.

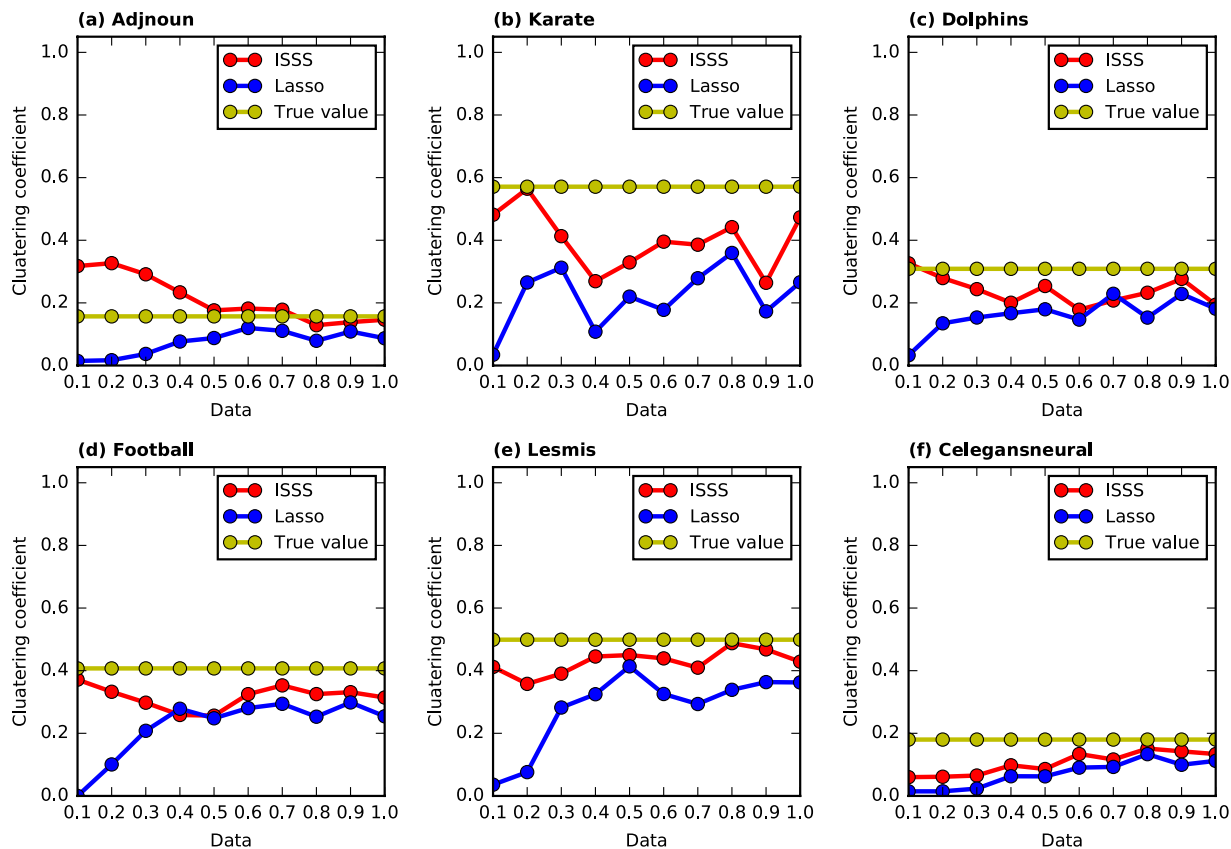


Figure 5. The clustering coefficient of the reconstructed network by ISSS and Lasso. The yellow line represents the real clustering coefficient of the network. The red line represents the clustering coefficient of the reconstructed network by ISSS.

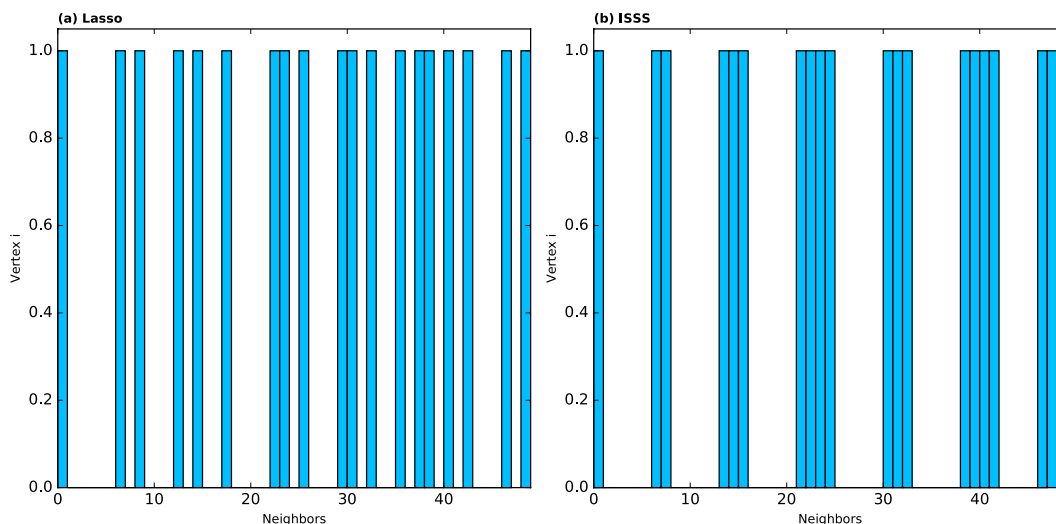


Figure 6. The reconstruction of vertex i 's neighbors. Suppose there are 50 vertices in all. If vertex i is connected with vertex j , the corresponding item in histogram is 1, otherwise it is 0. Lasso is good at predicting the neighbors in a discrete style, while the ISSS is inclined to predict continuous neighbors. The continuous neighbors are probably the way to construct the community structure of the network.

Discussion

In this paper, we propose the robust ISSS model in solving the network reconstruction problem. Compared with the state-of-art Lasso, the ISSS model is more robust, which can successfully infer the network structure with insufficient training data against noise. The sparse regularization term L_1 , L_2 and the structural TV penalties make

Data	0.05	0.10	0.15	0.20	0.4
ISSS	0.728	0.771	0.864	0.931	0.991
ISSS/TV	0.720	0.751	0.857	0.926	0.986
ISSS/L1	0.723	0.742	0.807	0.921	0.995
ISSS/L2	0.703	0.772	0.865	0.931	0.992
Lasso	0.542	0.587	0.677	0.757	0.941

Table 3. The performance of models with different penalties on a BA network. ISSS/TV is the ISSS model without TV penalty. ISSS/L1 is the ISSS model without L1 penalty. ISSS/L2 is the ISSS model without L2 penalty.

TV coefficient	Data				
	0.1	0.3	0.5	0.8	1
1.00E-06	0.8000	0.9700	0.9988	1.0000	1.0000
1.00E-05	0.8037	0.9750	0.9990	1.0000	1.0000
1.00E-04	0.7790	0.9815	0.9994	1.0000	1.0000
1.00E-03	0.7078	0.9830	0.9993	1.0000	1.0000
1.00E-02	0.7785	0.9538	0.9909	0.9996	0.9999
1.00E-01	0.7090	0.7832	0.8029	0.8363	0.8619
1.00E-00	0.4825	0.6137	0.6225	0.6406	0.6385

Table 4. Sensitivity analysis of TV penalty coefficient γ in Equation (4).

the ISSS model reconstruct the network with strong community structure. The experiments on the WS, ER and BA networks show that the ISSS outperforms Lasso even when the network system has very few observations with great noises. We also do some experiments to analyze the factors in identifying the performance of the ISSS, i.e. the variance of Φ , training data size and the clustering coefficient of the network.

In the future, on the one hand, we will continue to find out all the factors in identifying the upper and lower bound of ISSS's performance, and prove that from both theory and experiment perspectives. On the other hand, the community structure of a network defines the characteristic of a network, which identifies its functionality and should be preserved during the inferring process. Though ISSS model can identify the network with community structure, there could be room for improvement. For instance, the group Lasso and the tree group Lasso are the promising directions in network reconstruction task. Furthermore, we will optimize ISSS and apply the network inference method to solve the drug prediction and breast cancer prediction problems.

At present, it is still very challenging to infer many large-scale networks. Hence, it is a future trend with promising application field. And the evaluation criteria on directed and undirected networks are the same. As the development of network inference and compressed sensing research, more reasonable evaluation methods will be proposed to evaluate the experimental results on undirected networks.

Methods

The main goal of complex network reconstruction is to recover the connection \mathbf{X}_i by the payoff information y_i and virtual-payoff matrix Φ_i of vertex i from the time series of strategies. Note that the number of the neighbors of vertex i is often much less than the total number of vertices in the entire complex network, i.e., \mathbf{X}_i is sparse. Han *et al.*⁴⁰ adopted to use Lasso model to identify the neighbors of vertex i . By using Lasso model, the problem of network reconstruction is formulated as

$$\min_{\mathbf{X}_i} \|Y_i - \Phi_i \mathbf{X}_i\|_2^2 + \lambda \|\mathbf{X}_i\|_1, \quad (3)$$

where λ is a non-negative regularization parameter. All nonzero elements of the reconstructed \mathbf{X}_i correspond to the neighbors of i vertex. However, the Lasso-based model (3) only considers the sparse information.

In practice, to estimate a true signal in noise, the most frequently used methods are based on the total variation (TV) norm and the ℓ_1 norm. TV penalty forces sparsity on the spatial derivative of the weight map. TV norms are essential ℓ_1 norms of derivatives, hence ℓ_1 estimation procedures are more appropriate for the subject of TV estimation. The space of functions of bounded total variation plays an important role when accurate estimation of discontinuities in solutions is required²⁶. By contrast, total variation is remarkably effective at simultaneously preserving edges whilst smoothing away noise in flat regions for image denoising. Therefore, we consider to use ridge regression model with the TV regularization to preserve the structural information of networks and enhance the robustness performance of network reconstruction. After imposed the TV regularization, the objective of our method is formulated as

$$\min_{\mathbf{X}_i} F(\mathbf{X}_i) \triangleq \|Y_i - \Phi_i \mathbf{X}_i\|_2^2 + \frac{\beta}{2} \|\mathbf{X}_i\|_2^2 + \lambda \|\mathbf{X}_i\|_1 + \gamma TV(\mathbf{X}_i), \quad (4)$$

$$z_\mu^*(\mathbf{X}_i) = \operatorname{argmax}_{z \in \mathcal{K}} \left\{ \langle z, \mathbf{A}\mathbf{X}_i \rangle - \frac{\mu}{2} \|z\|_2^2 \right\}. \tag{8}$$

Theorem 1 (Nesterov’s Theorem²⁷). *Let the function s_μ be Nesterov’s smooth transform of a convex function TV penalty. We can obtain the following results:*

(1) s_μ is convex and differentiable with gradient $\nabla s_\mu(\mathbf{X}_i) = \mathbf{A}^T z_\mu^*(\mathbf{X}_i)$ and $\nabla s_\mu(\mathbf{X}_i)$ is Lipschitz continuous with constant $L_{s_\mu} = \|\mathbf{A}\|_2^2 / \mu$, where $\|\mathbf{A}\|_2^2 = \lambda_{\max}(\mathbf{A}^T \mathbf{A})$ is the largest eigenvalue of $\mathbf{A}^T \mathbf{A}$.

(2) For all $\mathbf{X}_i \in \mathbb{R}^p$, we have $\lim_{\mu \rightarrow 0} s_\mu(\mathbf{X}_i) = TV(\mathbf{X}_i)$.

The proof of Theorem 1 is obtained according to Lemma 1. According to the projection theorem⁴⁴, we know that $z_\mu^*(\mathbf{X}_i)$ is the projection of $\frac{1}{\mu} \mathbf{A}\mathbf{X}_i$ onto the compact and convex space \mathcal{K} , that is

$$z_\mu^*(\mathbf{X}_i) = P_{\mathcal{K}} \left(\frac{1}{\mu} \mathbf{A}\mathbf{X}_i \right) = \prod_{G \in \mathcal{G}} P_{\mathcal{K}_G} \left(\frac{1}{\mu} \mathbf{A}_G \mathbf{X}_i \right), \tag{9}$$

where \prod is the Cartesian product of a set and $P_{\mathcal{K}_G}(\cdot)$ is the projection onto each compact set \mathcal{K}_G defined as

$$P_{\mathcal{K}_G}(x) = \frac{x}{\max(\|x\|_2, 1)} = \begin{cases} x, & \text{if } \|x\|_2 \leq 1 \\ \frac{x}{\|x\|_2}, & \text{otherwise.} \end{cases}$$

Optimization. After smoothing by Nesterov’s technique, the proximal operator of s_μ becomes easily computed. We obtain a new optimization problem by using $s_\mu(\mathbf{X}_i)$ to replace the TV penalty of the problem (4):

$$\min_{\mathbf{X}_i} F_\mu(\mathbf{X}_i) \triangleq f_\mu(\mathbf{X}_i) + \lambda \|\mathbf{X}_i\|_1 = \frac{1}{2} \|y_i - \Phi_i \mathbf{X}_i\|_2^2 + \frac{\beta}{2} \|\mathbf{X}_i\|_2^2 + \lambda \|\mathbf{X}_i\|_1 + \gamma s_\mu(\mathbf{X}_i), \tag{10}$$

where $f_\mu(\mathbf{X}_i) = \|y_i - \Phi_i \mathbf{X}_i\|_2^2 + \frac{\beta}{2} \|\mathbf{X}_i\|_2^2 + \gamma s_\mu(\mathbf{X}_i)$ is a smooth convex function, i.e., continuously differentiable with Lipschitz continuous gradient constant L_{f_μ}

$$\|\nabla f_\mu(u) - \nabla f_\mu(v)\| \leq L_{f_\mu} \|u - v\|, \forall u, v \in \mathbb{R}^d,$$

and $s_\mu(\mathbf{X}_i)$ is Nesterov’s smoothing function of $TV(\mathbf{X}_i)$ and defined as Equation (7). Motivated by the idea of FISTA⁴², we adopt the following quadratic model as the approximation of $F_\mu(\mathbf{X}_i)$ at a given point v

$$Q_{L_{f_\mu}}(\mathbf{X}_i, v) = f_\mu(v) + \left\langle \mathbf{X}_i - v, \nabla f_\mu(v) \right\rangle + \frac{L_{f_\mu}}{2} \|\mathbf{X}_i - v\|^2 + \lambda \|\mathbf{X}_i\|_1, \tag{11}$$

which admits a unique minimizer

$$p_{L_{f_\mu}}(v) = \operatorname{argmin}_{\mathbf{X}_i} Q_L(\mathbf{X}_i, v) = \operatorname{argmin}_{\mathbf{X}_i} \left(\lambda \|\mathbf{X}_i\|_1 + \frac{1L_{f_\mu}}{2} \left\| \mathbf{X}_i - \left(v - \frac{1}{L_{f_\mu}} \nabla f(v) \right) \right\|^2 \right), \tag{12}$$

where $\nabla f(v) = \Phi_i^T (\Phi_i v - Y_i) + \beta v + \gamma \nabla s_\mu(v)$. Note that $p_{L_{f_\mu}}(v)$ is the well known component-wise soft thresholding operator

$$\left(p_{L_{f_\mu}}(v) \right)_i = \operatorname{sign}(u_i) (|u_i| - \lambda)_+,$$

where $u = v - \frac{1}{L_{f_\mu}} \nabla f(v)$ and $(\cdot)_+ = \max(\cdot, 0)$.

The main procedure of our method is described in Algorithm 1. Denote $t_0=0$, as FISTA, we generate two vectors $(v^{k+1}, \mathbf{X}_i^{k+1})$ to minimize the smoothed model in (10) at $k+1$ iteration:

$$v^{k+1} = \mathbf{X}_i^k + \frac{t_k - 1}{t_{k+1}} (\mathbf{X}_i^k - \mathbf{X}_i^{k-1}) \text{ and } \mathbf{X}_i^{k+1} = p_{L_{f_\mu}}(v^{k+1}), \tag{13}$$

where $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$. By using this iterative method to minimize the smoothed model in (10), it will converge to \mathbf{X}_i^* , which is the minimum of $F_\mu(\mathbf{X}_i)$. Moreover, $\|\mathbf{X}_i\|_1$ is a strong convex function, the convergence rate is guaranteed by

$$F_{\mu}(\mathbf{X}_i^k) - F(\mathbf{X}_i^*) \leq \frac{2L_{f_{\mu}} \|\mathbf{X}_i^0 - \mathbf{X}_i^*\|^2}{(k+1)^2}, \quad (14)$$

where k is the number of iterations. Meanwhile, by the duality gap theory and theoretical results in refs 27, 42, 48, we can obtain the following important result:

$$\lim_{k \rightarrow \infty} F_{\mu}(\mathbf{X}_i^k) = F(\mathbf{X}_i^*), \quad (15)$$

and the number of iterations is upper bounded by

$$\left(\frac{\sqrt{8\|\mathbf{A}\|_2^2 M \gamma^2}}{\varepsilon} + \frac{\sqrt{2L_{f_{\mu}}}}{\sqrt{\varepsilon}} \right) \|\mathbf{X}_i^0 - \mathbf{X}_i^*\|_2, \quad (16)$$

provided that $F_{\mu}(\mathbf{X}_i^k) - F(\mathbf{X}_i^*) < \varepsilon$. The more details of proof are referred to refs 27, 42 and 48.

Data Availability. All original data are available from the website FigShare https://figshare.com/articles/adjnoun_gml/4883708.

References

1. Ellison, N. B. *et al.* Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication* **13**, 210–230 (2007).
2. Zhang, J. & Philip, S. Y. Integrated anchor and social link predictions across social networks. *In IJCAI*, 2125–2132 (2015).
3. Du, W.-B. *et al.* Analysis of the chinese airline network as multi-layer networks. *Transportation Research Part E: Logistics and Transportation Review* **89**, 108–116 (2016).
4. Jeong, H., Mason, S. P., Barabási, A.-L. & Oltvai, Z. N. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
5. Zhang, J. *et al.* Learning entity types from query logs via graph-based modeling. *In Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, CIKM '15, 603–612 (ACM, New York, NY, USA, 2015).
6. Newman, M. E. A measure of betweenness centrality based on random walks. *Social networks* **27**, 39–54 (2005).
7. Okamoto, K., Chen, W. & Li, X.-Y. Ranking of closeness centrality for large-scale social networks. *In International Workshop on Frontiers in Algorithmics*, 186–195 (Springer, 2008).
8. Fortunato, S. Community detection in graphs. *Physics reports* **486**, 75–174 (2010).
9. Schneider, C. M., Moreira, A. A., Andrade, J. S., Havlin, S. & Herrmann, H. J. Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences* **108**, 3838–3841 (2011).
10. Dang, K., Shen, J.-Z., Dong, L.-D. & Xia, Y.-X. A graph route-based superframe scheduling scheme in wireless mesh networks for high robustness. *Wireless Personal Communications* **71**, 2431–2444 (2013).
11. Yang, Y., Li, Z., Chen, Y., Zhang, X. & Wang, S. Improving the robustness of complex networks with preserving community structure. *PloS one* **10**, e0116551 (2015).
12. Sun, P. G. Controllability and modularity of complex networks. *Information Sciences* **325**, 20–32 (2015).
13. Wang, Z., Wang, L., Szolnoki, A. & Perc, M. Evolutionary games on multilayer networks: a colloquium. *The European Physical Journal B* **88**, 1–15 (2015).
14. Wang, Z., Szolnoki, A. & Perc, M. Different perceptions of social dilemmas: Evolutionary multigames in structured populations. *Physical Review E* **90**, 032813 (2014).
15. Szolnoki, A. & Perc, M. Competition of tolerant strategies in the spatial public goods game. *New Journal of Physics* **18**, 083021 (2016).
16. Han, X., Shen, Z., Wang, W.-X. & Di, Z. Robust reconstruction of complex networks from sparse data. *Physical review letters* **114**, 028701 (2015).
17. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* **267**–288 (1996).
18. Hempel, S., Koseska, A., Kurths, J. & Nikołoski, Z. Inner composition alignment for inferring directed networks from short time series. *Physical review letters* **107**, 054101 (2011).
19. Gao, Y., Du, W. & Yan, G. Selectively-informed particle swarm optimization. *Scientific reports* **5**, 9295 (2015).
20. Du, W.-B., Gao, Y., Liu, C., Zheng, Z. & Wang, Z. Adequate is better: particle swarm optimization with limited-information. *Applied Mathematics and Computation* **268**, 832–838 (2015).
21. Gao, Z.-K. *et al.* Multi-frequency complex network from time series for uncovering oil-water flow structure. *Scientific reports* **5**, 8222 (2015).
22. Timme, M. Revealing network connectivity from response dynamics. *Physical review letters* **98**, 224101 (2007).
23. Napolitano, D. & Sauer, T. D. Reconstructing the topology of sparsely connected dynamical networks. *Physical Review E* **77**, 026103 (2008).
24. Marbach, D., Costello, J., Küffner, R., Vega, N. & Prill, R. Wisdom of crowds for robust gene network inference. *Nature* (2012).
25. Feizi, S., Marbach, D., Médard, M. & Kellis, M. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology* (2013).
26. Rudin, L. I., Osher, S. & Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* **60**, 259–268 (1992).
27. Nesterov, Y. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization* **16**, 235–249 (2005).
28. Erdős, P. & Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**, 43 (1960).
29. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *science* **286**, 509–512 (1999).
30. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *nature* **393**, 440–442 (1998).
31. Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal, Complex Systems* **1695**, 1–9 (2006).
32. Liu, W. & Lü, L. Link prediction based on local random walk. *EPL (Europhysics Letters)* **89**, 58007 (2010).
33. Sinatra, R. *et al.* The ultimatum game in complex networks. *Journal of Statistical Mechanics: Theory and Experiment* **2009**, P09012 (2009).
34. Han, X., Shen, Z., Wang, W., Lai, Y. & Grebogi, C. Reconstructing direct and indirect interactions in networked public goods game. *Sci Reports* **6**, 30241 (2016).
35. Albert, R. Network inference, analysis, and modeling in systems biology. *The Plant Cell* **19**, 3327–3338 (2007).

36. Wang, H., Qian, L. & Dougherty, E. Inference of gene regulatory networks using s-system: a unified approach. *IET systems biology* **4**, 145–156 (2010).
37. Ali, R., Kim, S. W., Kim, B.-S. & Park, Y. Design of mac layer resource allocation schemes for ieee 802.11 ax: Future directions. *IETE Technical Review* 1–25 (2016).
38. Zahoranszky-Kohalmi, G. *Network inference driven drug discovery*. Ph.D. thesis, The University of New Mexico (2016).
39. Bewick, V., Cheek, L. & Ball, J. Statistics review 13: Receiver operating characteristic curves. *Critical Care* **8**, 508–512 (2004).
40. Han, X., Shen, Z., Wang, W. & Di, Z. Robust reconstruction of complex networks from sparse data. *Phys Rev Lett* **114**, 028701 (2015).
41. Daubechies, I., Defrise, M. & De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics* **57**, 1413–1457 (2004).
42. Beck, A. & Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* **2**, 183–202 (2009).
43. Chambolle, A. & Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* **40**, 120–145 (2011).
44. Bertsekas, D. P. *Nonlinear programming* (Athena scientific Belmont, 1999).
45. Liu, D. C. & Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical programming* **45**, 503–528 (1989).
46. Li, D.-H. & Fukushima, M. A modified bfgs method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics* **129**, 15–35 (2001).
47. Bonnans, J.-F., Gilbert, J. C., Lemaréchal, C. & Sagastizábal, C. A. *Numerical optimization: theoretical and practical aspects* (Springer Science & Business Media, 2006).
48. Hadj-Sellem, F., Lofstedt, T., Frouin, V., Guillemot, V. & Duchesnay, E. An iterative smoothing algorithm for regression with structured sparsity. *arXiv preprint arXiv:1605.09658* (2016).

Acknowledgements

We thank Wenbo Du for useful discussions. We also thank Mark Newman and Xiao Han for providing the point of presence data. This work was supported by the National Natural Science Foundation of China (Grand Nos 61370126, 61672081, U1636211), National High Technology Research and Development Program of China (No. 2015AA016004), Beijing Advanced Innovation Center for Imaging Technology (No. BAICIT-2016001), the Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE-2015ZX-16).

Author Contributions

Y.Y. and Z.J.L. developed the study concept and design. Y.Y. collected the data. Y.Y. and T.J.L. analyzed the data. Y.Y. and X.M.Z. interpreted the data, and Y.Y., T.J.L. and P.S.Y. drafted the manuscript. All authors provided critical revisions and approved the final version. The first two authors contribute equal contributions to this paper.

Additional Information

Competing Interests: The authors declare that they have no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2017