Self-Guided Community Detection on Networks with Missing Edges

Dongxiao He[1,†], **Shuai Li**[1,†], **Di Jin**[1], **Pengfei Jiao**[2,*] and **Yuxiao Huang**[3]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Center of Biosafety Research and Strategy, Law School, Tianjin University, Tianjin, China
[3]Data Science, George Washington University, Washington, D.C., USA
{hedongxiao, dedao, pjiao, jindi}@tju.edu.cn, yuxiaohuang@gwu.edu

Abstract

The vast majority of community detection algorithms assume that the networks are totally observed. However, in reality many networks cannot be fully observed. On such network is edges-missing network, where some relationships (edges) between two entities are missing. Recently, several works have been proposed to solve this problem by combining link prediction and community detection in a two-stage method or in a unified framework. However, the goal of link prediction, which is to predict as many correct edges as possible, is not consistent with the requirement that is predicting important edges for identifying communities on edges-missing networks. Thus, combining link prediction and community detection cannot work very well in terms of detecting community structure for edges-missing networks. In this paper, we propose a community self-guided generative model which jointly completes the edges-missing network and identifies communities. In our new model, completing missing edges and identifying communities are not isolated but closely intertwined. Furthermore, we developed an effective model inference method that combines a nested Expectation-Maximization algorithm and Metropolis-Hastings sampling. Extensive experiments on real-world edges-missing networks show that our model can effectively detect community structures while completing missing edges.

1 Introduction

With the development of computer technology and the massive increase in data, network has become an important and ubiquitous structure in real world, which can describe relationship (represented by edges) between entities (represented by nodes). Such networks include, for example, social networks, biological networks and citation networks, just to name a few [Barrat *et al.*, 2008; Cohen and Havlin, 2010; Newman, 2018]. Community detection, one of the most important tasks in network analysis, has attracted great attention. In general, the goal of community detection is to discover functionally related modules which generally has the following structural characteristics: nodes in the same community are densely connected whereas nodes in different communities are sparsely connected [Girvan and Newman, 2002]. Community detection has been successfully applied to a wide range of areas. For instance, social networks allow us to recommend advertisements to users with similar hobbies; Protein-protein interaction networks permit us to discover specific functional modules of proteins. Many community detection algorithms have been proposed. They include, for example, modularity optimization based methods [Newman and Girvan, 2004; MacMahon and Garlaschelli, 2015], deep learning based methods [Rozemberczki *et al.*, 2018; Wang *et al.*, 2017; Li *et al.*, 2020] and stochastic block models (SBM) [Ball *et al.*, 2011; Peixoto, 2014; Karrer and Newman, 2011; Yang and Leskovec, 2013; Zhang *et al.*, 2015; Ye *et al.*, 2018].

These algorithms mentioned above assume that the observed networks are complete. However, there are missing edges in many real-world networks. For instance, some users on Twitter may hide part of their friends list, making the edge between the corresponding pair of users unobserved in the network [Lin *et al.*, 2013]. In the terrorist organization network [Lin *et al.*, 2013], where each node represents terrorist activities, and the edge between two nodes indicates that two activities come from a same organization, we may not know which terrorist organization carried out a terrorist activity. Thus, the relationships between some terrorist activities are unknown. Hereafter network with missing edges is called edges-missing network. As the topology information plays an important role in community detection, intuitively using traditional algorithms designed for complete networks directly on edges-missing networks could compromise community detection accuracy. This is verified by Gabielkov [2012], where various different community detection algorithms were tested on many edges-missing networks. In addition, missing edges have been shown to have impact on some important network properties, such as the diameter, the centrality or the degree distribution [Huisman, 2009; Borgatti *et al.*, 2006]. Therefore, it is necessary to design community detection algorithm that is particularly suitable for edges-missing networks.

*Corresponding author.
†Contributed equally to this work.

Recently, some community detection algorithms for edges-missing networks have been proposed. These methods fall into two categories. The first category includes two-stage algorithms [Tran *et al.*, 2018]. Concretely, the first stage entails performing link prediction to recover the missing edges, and the second stage entails performing community detection on the complemented network. The second category includes methods conducting community detection and link prediction simultaneously in a unified framework [Shao *et al.*, 2019]. Such as CLMC [Shao *et al.*, 2019], is one of the most representative algorithms in this category, whose goal is to learn a similarity matrix to detect communities and a supplement matrix to predict missing edges simultaneously in a unified framework. It is worth noting that all these algorithms assume that the more missing edges the algorithm can predict correctly, the more accurate the community detection is. However, the assumption may not hold in reality as not all edges play the same role in community detection. For some nodes, even several edges are missing, their community memberships are still clear. Conversely, for other nodes, even one missing edge could change their community membership. For example, we applied this representative unified method CLMC on the classical community detection dataset, Zachary's Karate club network, and we removed 10% edges randomly from the network, making it an edges-missing network. The result is shown in Figure 1 (a), where although most of edges are predicted correctly, the results of community detection are still biased, i.e., node 2 is divided into a wrong community. This is because the correctly predicted edges do not play an important role in community detection, echoing what we illustratively explained earlier. For instance, removing edge (1, 17) (which is correctly predicted) will not change the community membership of node 1 and node 17. This is due to the fact that the true community of node 1 and 17 are easy to determine, and most of their neighbors only belong to one community and they have few connections to another community. This is also the case for edges (29, 33) and (1, 7). On the contrary, node 2 is located at the boundary of two communities, which is difficult to determine its community membership. For this node, the algorithm CLMC did not predict its missing edges which indirectly led it being placed into the wrong community. In fact, link prediction algorithms prefer to predict the intra-community edges that play less important role in community detection. Thus, simple combination of link prediction and community detection cannot handle the problem of community detection on edges-missing networks in essential.

To address this problem, we proposed a community self-guided generative model for jointly completing the edges-missing network and identifying communities. Unlike the state-of-the-arts such as CLMC, our model predicted the missing edges (2, 3) correctly and divided node 2 into the correct community (as shown in Figure 1 (b)). The improvement is brought about by accommodating two sets of variables in our model, one for recovering the missing edges, and one for characterizing communities. In this new model, recovering missing edges is guided by community detection in such a way that we predict the edges that are important for identifying communities. Meanwhile, community detection is also affected by completing the network. For training the model, we developed an effective method that combines a nested Expectation-Maximization (EM) algorithm and Metropolis-Hastings sampling. We evaluated our model on a variety of real-world networks and compared with different kinds of methods, including methods for complete networks, two-stage methods and unified end-to-end methods for edges-missing networks. Empirical results show the significant superiority of a model over the existing methods.

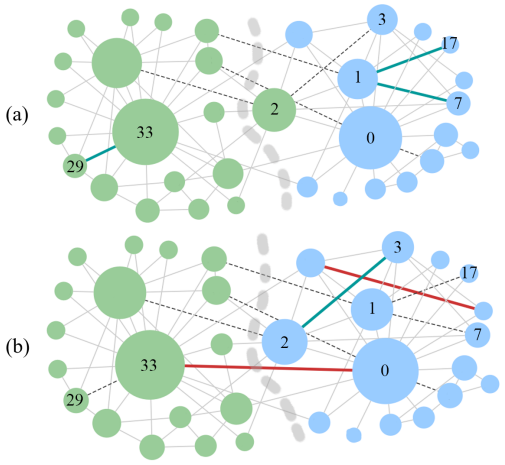


Figure 1: An motivated example on the Zachary's Karate club network. The bold dashed line indicates the split observed in real world. These two colors on nodes, blue and green, represent two communities inferred by community detection algorithms. The black dashed line represents missing edge. The red and green lines indicate the wrong and correct edges inferred by algorithms. We remove 10% edges randomly, and then apply CLMC and our algorithm to perform (a) and (b) results.

2 The Community Self-Guided Approach

In this paper, we developed a community self-guided generative model for completing the edges-missing network and discovering community structure at the same time. To train the model, we developed an effective method for combining a nested Expectation-Maximization (EM) algorithm and Metropolis-Hastings sampling approach.

2.1 The Model

Considering an observed network $G_O\,(V, E_O)$ with n nodes and $|E_O|$ edges. We use $G_M\,(V, E_M)$ to represent a latent network $G_M\,(V, E_M)$, where E_M denotes the missing edges of the observed network G_O. We define the complete graph containing the observed network G_O and the latent network G_M as $G(V, E)$, where $E = E_O \cup E_M$. We use $A = (a_{ij})_{n\times n}$ to denote the adjacent matrix of the observed network G_O, where $a_{ij} = 1$ if an edge exists between node i and node j, or 0 otherwise. We use $Z = (z_{ij})_{n\times n}$ to denote the adjacent matrix of the latent network G_M, where $z_{ij} = 1$ if the edge between node i and node j is missing, or 0 otherwise. Then, with a community assumption, complete network G (where $G = G_O \cup G_M$) can be viewed as an ensemble of c *probabilistic* communities, $\{G_1, G_2, \ldots G_c\}$, where every

node has a probabilistic membership in each community G_k, $k = 1, 2 \cdots c$. For each community, it can be regarded as a random graph with no community structure, and we use a random-graph null model to model each probabilistic community as Jin [2015]. Note that the community memberships are designed for the complete network including both the observed part and the missing part.

Now we give the specific definition of the model. We use d_{ik} to denote the expected degree of node i within community G_k. As every probabilistic community G_k is a random graph, the expected number of edges between node i and node j in community G_k is:

$$\widehat{w}_{ij}^{k} = \frac{d_{ik} d_{jk}}{\sum_r d_{rk}} \tag{1}$$

Taking all *k* probabilistic communities into account, the expected number of edges between node i and node j can be defined as:

$$\widehat{w}_{ij} = \sum_k \widehat{w}_{ij}^{k} = \sum_k \frac{d_{ik} d_{jk}}{\sum_r d_{rk}} \tag{2}$$

The exact number of edges between nodes i and j is Poisson distributed about this mean value $\widehat{w}_{ij}$. Then, the likelihood function that the complete network G (including the observed network G_o and the latent network G_M) was presumably generated from the model can be defined as follow:

$$\begin{aligned} &P(A, Z \mid D) \\ &= \prod_{i,j \in (A \cup Z)} \frac{\left(\sum_k \frac{d_{ik} d_{jk}}{\sum_r d_{rk}} \right)^{w_{ij}}}{w_{ij}!} \exp\left(- \sum_k \frac{d_{ik} d_{jk}}{\sum_r d_{rk}} \right) \end{aligned} \tag{3}$$

where A denotes the adjacent matrix of the observed network G_O, Z the adjacent matrix of the latent network G_M, and $D = (d_{ik})_{n \times c}$ the expected degrees of nodes for all communities. In fact, the latent network G_M is not known, and the missing edges in Z can be considered as latent variables.

In this case, our model integrates the missing edges and the community memberships in a unified generative model. With this configuration of Z, it can help to produce the complete network which helps the community detection best. Then, the latent network and community structures can be learned simultaneously and are mutual enhancement. By doing so, the model has better performance than the two-stage methods. More importantly, compared to previous unified frameworks, such as CLMC, our model focuses on the important nodes and those from community boundary based on the parameter D. Furthermore, by calculating the subordination of nodes degree participating in communities, our model can recognize latent edges that are the most important for community detection. To sum up, our model is self-guided by community structure and designed for missing edge networks.

In our model, the model parameters d_{ik} is the community information and they are also used to completing the missing part. Given d_{ik}, the expected number of edges $\widehat{w}_{ij}$ between any node i and node j can be obtained, and then the probability of the existence of missing edge between any unconnected pair nodes of the observed network G_O can be modeled by Bernoulli distribution with parameter $\widehat{w}_{ij}$. The completing missing edges and the discovering community structures are closely intertwined. Once the missing part Z is obtained, the model parameters d_{ik} can be estimated by maximizing the likelihood function. After the model parameters d_{ik} are estimated, the configuration of the missing part that results in the largest increase of the likelihood function is selected. The selected configuration will result in a new potential maximum value of the likelihood function. Then the model parameters d_{ik} is refined by finding the optimal value corresponding to the new potential maximum value. As we keep alternating between estimating the missing part Z and model parameters d_{ik}, the community structure becomes clearer and clearer.

Although the determination of the missing part Z has been modeled by Bernoulli distribution with the expected number of edges as its parameters, the sample Z that is directly generated by Bernoulli distribution is highly random. In order to minimize the impact of randomness on likelihood and generate a confident sample Z, we develop a Metropolis-Hastings sampling process with a Markov chain. According to Markov property, given sampled $Z^{(t)}$, the next candidate sample $Z^{(t+1)}$ can be generated. Then we accept or reject this new $Z^{(t+1)}$ according to an acceptance probability. This process continues until reaching a steady state. According to the mechanism of Metropolis-Hastings sampling, when it reaches a stationary state, the sampled samples can effectively improve the likelihood, and the determined missing part can improve the performance of community detection.

2.2 The Model Inference

We then describe how to infer community structure and the latent network simultaneously on edges-missing networks by using a nested Expectation-Maximization (EM) algorithm with Metropolis-Hastings sampling.

E-Step with Metropolis-Hastings Sampling

Given the model parameters D, we use Metropolis-Hastings sampling to determine the missing part Z from the Bernoulli distribution with parameter $\widehat{W} = (\widehat{w}_{ij})_{n \times n}$, where $\widehat{w}_{ij}$ is the expected number of edges between node i and node j. To make the sampling process effective, we adopted the strategy of [Kim and Leskovec, 2011] which can produce a sample of Z in constant time. According to Markov property, given the current sample $Z^{(t)}$, the next candidate sample $\tilde{Z}$ can be generated. The new sample $\tilde{Z}$ is generated based on $Z^{(t)}$ by removing one edge randomly from $Z^{(t)}$ and adding another edge to it as [Kim and Leskovec, 2011]. This sample $\tilde{Z}$ is accepted as the next sample $Z^{(t+1)}$ with an acceptance probability. If the new sample $Z^{(t+1)}$ is rejected, the current sample $Z^{(t)}$ is taken as the next sample $Z^{(t+1)}$. The acceptance rate $R\left(\tilde{Z}, Z^{(t)}\right)$ is defined as:

$$R\left(\tilde{Z}, Z^{(t)}\right) = \min\left(1, \frac{P(\tilde{Z}) P\left(Z^{(t)} \mid \tilde{Z}\right)}{P\left(Z^{(t)}\right) P\left(\widetilde{Z} \mid Z^{(t)}\right)}\right) \tag{4}$$

where the transition probability $P\left(Z^{(t)}\right) P\left(\widetilde{Z} \mid Z^{(t)}\right)$ can be

defined as:

$$\begin{aligned} &P\left(Z^{(t)}\right) P\left(\widetilde{Z} \mid Z^{(t)}\right) \\ &= P\left(Z^{(t)} \backslash e_1, e_2\right) P\left(e_1 \in E_M^{(t)}\right) P\left(e_2 \notin E_M^{(t)}\right) P(\text{del } e_1, \text{add } e_2) \\ &= P\left(Z^{(t)} \backslash e_1, e_2\right) P(e_1)(1-P(e_2)) \frac{1}{\left|E_M^{(t)}\right|} \frac{P\left(e_2\right)}{\sum_{e \notin E_M^{(t)}} P(e)+P(e_1)} \end{aligned} \tag{5}$$

where $Z^{(t)} \backslash e_1, e_2$ means edges e_1 and e_2 are not in $Z^{(t)}$, and $E_M^{(t)}$ denotes the set of edges in sample $Z^{(t)}$. Similarly, $P(\tilde{Z}) P\left(Z^{(t)} \mid \tilde{Z}\right)$ is defined as:

$$\begin{aligned} &P(\tilde{Z}) P\left(Z^{(t)} \mid \tilde{Z}\right) \\ &= P\left(\tilde{Z} \backslash e_1, e_2\right) P(e_2)(1-P(e_1)) \frac{1}{\left|\tilde{E}_M\right|} \frac{P\left(e_1\right)}{\sum_{e \notin \tilde{E}_M} P(e)+P(e_2)} \end{aligned} \tag{6}$$

The final acceptance probability can be derived as follows:

$$R\left(\tilde{Z}, Z^{(t)}\right) = \min\left(1, \frac{1-P(e_1)}{1-P(e_2)}\right) \tag{7}$$

Using the above acceptance probability, we generate samples of Z and get a mean matrix $\bar{F}$ as follows:

$$\bar{F} = \left(Z^{(W)}, Z^{(W+1)}, \ldots Z^{(W+S)}\right) / S \tag{8}$$

where W is the warm-up iterations and S is the total number of samples. $\bar{F}$ is an $n \times n$ matrix, where each element f_{ij} in $\bar{F}$ represents the strength between node i and node j. The confident sample Z can be derived through $\bar{F}$ by setting a threshold value α:

$$\begin{aligned} Z_{\text{confident}} = &\{z_{ij} = 1 \text{ when } f_{ij} \geq \alpha; \\ &\text{otherwise } z_{ij} = 0 \mid z_{ij} \in Z, f_{ij} \in \bar{F}\} \end{aligned} \tag{9}$$

M-Step with Nested EM

In M-step, we consider the missing part Z and the observed part A are given, and the goal is to estimate the model parameters D that maximize the likelihood function in (3), that is, finding the parameters that fit the model to the given data (the missing part Z and the observed part A) best. Here maximizing the likelihood is typically done by maximizing the logarithm of the likelihood, which makes the differentiation much simpler while giving the exact same result. Taking the logarithm of (3), we can get the log-likelihood:

$$L = \sum_{ij} w_{ij} \log\left(\sum_k \frac{d_{ik} d_{jk}}{\sum_r d_{rk}}\right) - \sum_{ijk}\left(\sum_k \frac{d_{ik} d_{jk}}{\sum_r d_{rk}}\right) \tag{10}$$

Since directly maximizing (10) by differentiation will lead to a set of nonlinear implicit equations for the model parameters d_{ik}, here we apply an EM algorithm to solve this problem. Applying Jensen's inequality to (10), we get:

$$\begin{aligned} \bar{L}\left(d_{ik}, q_{ij,k}\right) = &\sum_{ijk}\left(w_{ij} q_{ij,k} \log \frac{d_{ik} d_{jk} / \sum_r d_{rk}}{q_{ij,k}}\right) \\ &- \sum_{ijk}\left(\frac{d_{ik} d_{jk}}{\sum_r d_{rk}}\right) \leq L\left(d_{jk}\right) \end{aligned} \tag{11}$$

where $\bar{L}$ is a lower bound of L and the probabilities $q_{ij,k}$ can be freely chosen provided satisfy $\sum_k q_{ij,k} = 1$. And the equality holds at:

$$q_{ij,k} = \frac{d_{ik} d_{jk} / \sum_r d_{rk}}{\sum_l \left(d_{il} d_{jl} / \sum_r d_{rl}\right)} \tag{12}$$

Maximizing the log-likelihood L with respect to the model parameters d_{ik} equates maximizing its lower bound $\bar{L}\left(d_{ik}, q_{ij,k}\right)$ with respect to both the model parameters d_{ik} and the probabilities $q_{ij,k}$.The EM algorithm for this dual maximization is to iteratively maximize $q_{ij,k}$(i.e., the E-step) and then maximize the model parameters d_{ik}(i.e., the M-step),which has been proved to converge to a local maximum.

For the E-step, we need to make $\bar{L} = L$. Given the optional model parameters d_{ik},the optimal values of $q_{ij,Z}$ can be obtained by Eq.(12); For the M-step, given the optional values of $q_{ij,k}$, the optional d_{ik} can be found by differentiating (12) under the constraints $\sum_k d_{ik} = d_i$, which gives:

$$d_{ik} = \sum_j w_{ij} q_{ij,k} \tag{13}$$

3 Experiments

We first compare our method SGCD with some state-of-the-art methods on real networks with ground-truth communities, and then on networks without known communities. Next, we give a case study analysis to show why our method works and, moreover, why it is superior to the link prediction enhanced end-to-end strategies. We finally test its effectiveness by varying the radio of missing edges, as well as its stability on the unique hyperparameter α.

3.1 On Networks with Known Communities

We test the performance of our method SGCD on six widely-used real networks with known communities (Table 1). Since these networks are fully observed, according to [Shao *et al.*, 2019], we randomly removed 20% existing edges for each network to produce edges-missing networks. To be specific, for each network we generated 50 edges-missing network instances randomly, and calculated the mean of the performance of each algorithm tested on these networks.

We compare SGCD with some most related methods (Table 2). BigClam [Yang and Leskovec, 2013], DANMF [Ye *et al.*, 2018] and GEMSEC [Rozemberczki *et al.*, 2018] are traditional community detection methods which do not consider missing edges. NM-SBM [Peixoto, 2018] and CLMC [Shao *et al.*, 2019] are designed for finding communities on edges-missing networks. MNDP [Jin *et al.*, 2015] is the base of our method not considering missing edges, $\text{MNDP}_{\text{2Stage}}$ is the two-stage version of MNDP considering missing edges (i.e., we use the results of the first run of MNDP to fill up missing edges for its second run), and $\text{MNDP}_{\text{Full}}$ is MNDP running on the original complete network without removing edges. $\text{SGCD}_{\text{Full}}$ is our SGCD running on the complete network. We use the default settings for all the baselines, and set α= 0.5 for our method (see the parameter analysis section later). We run each method 20 times on each network instance and select the result corresponding to the maximum

likelihood since they are mainly model-based methods. We use Normalized Mutual Information (NMI) as the measure metric as it is widely-used in community detection.

Datasets	# nodes	# edges	# classes
Zachary's karate club	34	78	2
Dolphin social network	62	160	2
Political books	105	441	3
Football	115	613	12
Political blogs	1,490	16,717	2
Pubmed	19,717	44,338	3

Table 1: Real-world networks with ground-truth communities.

The results are shown in Table 2. As shown, our SGCD performs the best on all the networks with missing edges, and is very close to the gold standard, i.e., $\text{MNDP}_{\text{Full}}$. Particularl, our SGCD outperforms 7.39% on average over the second-best method CLMC (which is also an end-to-end model but uses link prediction to reinforce community detection). In addition, thanks to the effectiveness of this new self-guided mechanism, $\text{SGCD}_{\text{Full}}$ can even improve $\text{MNDP}_{\text{Full}}$ which can be taken as an upper bound of the link prediction-enhanced methods like CLMC. This further shows that, SGCD not only has the superiority on networks with missing edges, but can also improve community detection performance on full networks. This may further explain why it is more effective than the link prediction based community detection for edge-missing networks.

3.2 On Networks without Known Communities

We further test SGCD on several real networks without known communities (Table 3). Because the model-based methods compared need the number of communities, we derive it by using the well-known Louvain method, as suggested by [Newman and Girvan, 2004]. While this number may not be completely correct, it does not affect the fairness of the evaluation and comparison of these methods in general. We use modularity Q as the evaluation metric which is often the most widely-used metric in this case.

The results are shown in Table 4. As shown, SGCD performs best on 3 out of the 5 networks, while CLMC performs best on the remaining two networks. But on average, SGCD improves CLMC 0.1791 under the condition that Q is typically in the range of 0.2 to 0.8. Furthermore, on the complete networks, $\text{SGCD}_{\text{Full}}$ improves $\text{MNDP}_{\text{Full}}$ 0.0906 on average. This further validates the effectiveness of the new framework with our self-guided mechanism.

Datasets	# nodes	# edges	# classes
Les Miserables	77	254	6
Word adjacencies	112	425	7
Jazz musicians collaborations	198	2,742	4
C. Elegans neural	297	2,148	5
E-mail network URV	1,133	5,451	11

Table 3: Real-world networks with unknown community structures.

Metrics (%)	Methods	Karate	Dolphin	Polbooks	Football	Polblogs	Pubmed
NMI	BigClam	71.01	59.33	30.87	60.51	0.93	1.21
	DANMF	73.75	74.37	52.71	88.69	51.71	7.34
	GEMSEC	73.08	73.53	49.85	84.90	42.69	6.47
	NM-SBM	64.38	51.02	48.15	88.47	41.63	5.37
	CLMC	79.66	82.61	52.54	88.06	17.98	3.29
	MNDP	71.37	70.13	45.05	85.73	45.14	6.53
	MNDP_{2S}	72.08	77.34	48.30	86.20	47.26	5.79
	SGCD	**82.72**	**83.70**	**53.63**	**88.90**	**52.07**	**7.49**
	$\text{MNDP}_{\text{Full}}$	**100**	**88.88**	53.98	92.54	54.73	10.30
	$\text{SGCD}_{\text{Full}}$	**100**	**88.88**	**56.43**	**92.63**	**55.30**	**10.70**

Table 2: Comparison of different methods in terms of NMI. Each result is averaged on 50 randomly generate edges-missing network instances for each network by removing 20% edges. Differently, $\text{MNDP}_{\text{Full}}$ and $\text{SGCD}_{\text{Full}}$ show results on the complete network without removing edges.

Metrics	Methods	Les	Word	Jazz	C,Elegans	E-mail
Modularity Q	BigClam	0.4217	0.0509	0.2649	0.0456	0.0400
	DANMF	0.3464	0.0370	0.2888	0.2435	0.4954
	GEMSEC	0.5093	0.2563	0.4124	0.3607	0.4756
	NM-SBM	0.4760	0.2507	0.4034	0.3523	0.4931
	CLMC	0.1988	0.0973	**0.4885**	**0.3788**	0.0216
	MNDP	0.5207	0.2658	0.3915	0.3685	0.4758
	MNDP_{2S}	0.5314	0.2824	0.4215	0.3674	0.5001
	SGCD	**0.5355**	**0.2832**	0.3962	0.3648	**0.5011**
	$\text{MNDP}_{\text{Full}}$	0.5426	0.3343	0.4405	0.4029	0.5517
	$\text{SGCD}_{\text{Full}}$	**0.5680**	**0.3354**	**0.4473**	**0.4082**	**0.5584**

Table 4: Comparison of different methods on networks without ground-truth in terms of modularity Q.

3.3 Why SGCD Works

We give an illustrative experiment to show why our method SGCD works on networks with missing edges. Here we select two representative algorithms MNDP and CLMC to be compared, and use the dolphin social network with 20% missing edges as a sample dataset. The results of MNDP, CLMC and SGCD are shown in Figure 3 (a), (b) and (c) respectively. Compared to MNDP, our SGCD divides nodes 28 and 39 into correct communities by producing important edges, such as edges (28, 36) and (28, 39). Compared to CLMC, our SGCD divides node 39 into the correct community. In this case, while CLMC predicts many edges correctly, as most of these edges do not play an important role in community detection, the result is still compromised. Conversely, thanks to the effective self-guided mechanism, our SGCD produces edges like (28, 39) that significantly improve community structures. This could be the reason why our new framework is effective for networks with missing edges, and better than other types of methods (e,g., CLMC) which allow missing edges.

3.4 Varying the Ratio of Missing Edges

The goal here is to show the robustness of SGCD on networks with varying the ratios of missing edges. Here we only show the results on two sample networks due to space limitation as other networks are similar. For each network, we chose eight different edges-missing ratios, i.e., from 0% to 35% with step size of 5%. (We did not consider the ratio higher than 40% because it will make the network too sparse to be detected, according to the theoretical limits of detectability [Decelle *et al.*, 2011]) The results are shown in

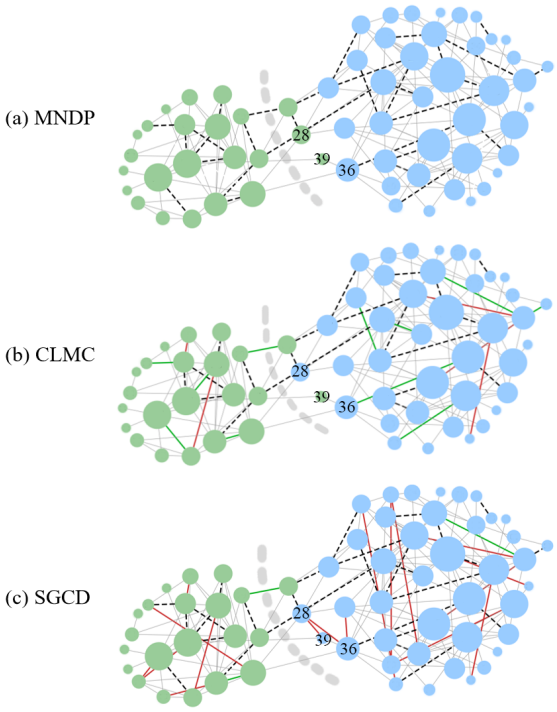
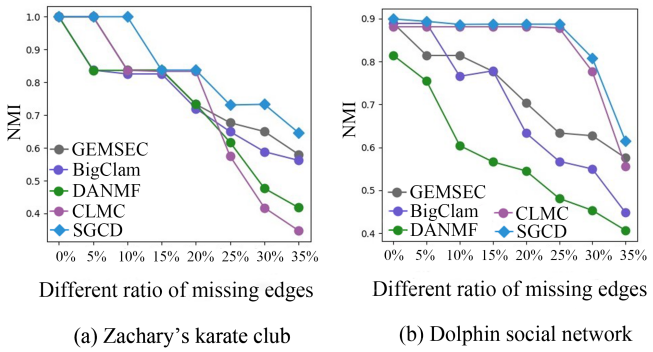
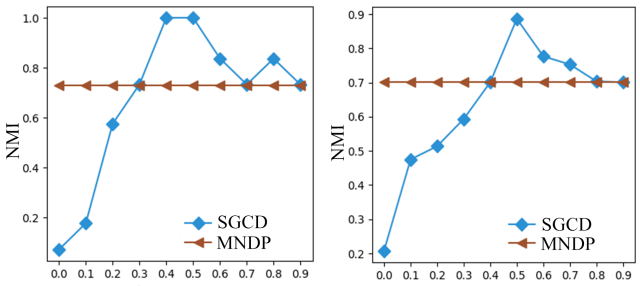


Figure 2: Comparison of (a) MNDP, (b) CLMC and (c) SGCD on the dolphin social network with 20% edges removed as an example. Different colors of nodes denote different communities inferred by the algorithm, and real communities are splatted by Bold grey dashed lines. Grey edges represent the existing edges, black dashed edges represent the removed edges, and red (and blue) edges the incorrectly (and correctly) predicted edges.



(a) Zachary's karate club (b) Dolphin social network

Figure 3: Performance of different methods on networks with different ratio of missing edges.



(a) Zachary's karate club (b) Dolphin social network

Figure 4: The performace of SGCD with varying the values of hyperparameters on networks. We take the results of MNDP as the baseline because it is the base of SGCD.

Figure 3, where our model achieves the best performance in most cases. Even when the ratio is 0%, which means MNDP reduces to $\text{MNDP}_{\text{Full}}$, our SGCD still performs better. This further demonstrates the advantages of our new model due to this community self-guided mechanism.

3.5 Parameter Analysis

Our model has only one hyperparameter α which is to control the threshold of the probability of filling out edges. We select the same networks as those in the above section as examples to test the sensibility our SGCD on α. As shown in Figure 4, when α approaches 0.0, the result of SGCD was inaccurate compared to MNDP. This is because the lower the α, the more noise edges are added. As a result, these noise edges may produce negative effects on inferring community structures based on the observed edges. When α approaches 0.9, which leads to very few edges added to observed data, the NMI accuracy of SGCD and MNDP are similar. However, when α is in the range of 0.4 to 0.5, SGCD is often stable and gives better results and this trend is similar on other networks. Therefore, we set $\alpha = 0.5$ without loss of generality.

4 Conclusion and Discussion

In this paper, we studied the problem of community detection on edges-missing networks, and proposed a community self-guided generative model SGCD. The model jointly formulates missing edges and community memberships in a unified likelihood function, which can realize predicting important edges for community detection by choosing edges that increase likelihood mostly. We developed an efficient inference method by using a nested EM algorithm with Metropolis-Hastings Sampling to train the model. We finally tested our SGCD on eleven real-world edges-missing networks with or without ground-truth, and compared with several state-of-the-art approaches. The results showed that this new approach outperformed all the methods compared.

The new model is still not perfect. That is, when the ratio of missing edges reaches a certain level, it will hard to find edges so effective to help community detection. While there are several works working on it, it is still a bank for edges-missing networks which we will take as the future work.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (61876128, 61832014, 61772361, 61902278), the Tianjin Municipal Science and Technology Project (Grant No. 19ZXZNGX00030) and University Facilitating Fund at Geroge Washington University.

References

[Ball *et al.*, 2011] Brian Ball, Brian Karrer, and M. E. J. Newman. Efficient and principled method for detecting communities in networks. *Physical Review E*, 84(3):036103, Sep 2011.

[Barrat *et al.*, 2008] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, Cambridge, 2008.

[Borgatti *et al.*, 2006] Stephen P Borgatti, Kathleen M Carley, and David Krackhardt. On the robustness of centrality measures under conditions of imperfect data. *Social networks*, 28(2):124–136, October 2006.

[Cohen and Havlin, 2010] Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge university press, Cambridge, 2010.

[Decelle *et al.*, 2011] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Inference and phase transitions in the detection of modules in sparse networks. *Physical Review Letters*, 107(6):065701, August 2011.

[Gabielkov and Legout, 2012] Maksym Gabielkov and Arnaud Legout. The complete picture of the twitter social graph. In *Proceedings of the 2012 ACM conference on CoNEXT student workshop*, pages 19–20, Nice, France, December 2012. ACM.

[Girvan and Newman, 2002] Michelle Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

[Huisman, 2009] Mark Huisman. Imputation of missing network data: Some simple procedures. *Journal of Social Structure*, 10(1):1–29, January 2009.

[Jin *et al.*, 2015] Di Jin, Zheng Chen, Dongxiao He, and Weixiong Zhang. Modeling with node degree preservation can accurately find communities. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 160–167, Austin, Texas, January 2015. AAAI Press.

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

[Kim and Leskovec, 2011] Myunghwan Kim and Jure Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of the 11th SIAM International Conference on Data Mining*, pages 47–58, Arizona, USA, April 2011. Society for Industrial and Applied Mathematics.

[Li *et al.*, 2020] Jia Li, Honglei Zhang, Zhichao Han, Yu Rong, Hong Cheng, and Junzhou Huang. Adversarial attack on community detection by hiding individuals. In *Proceedings of The Web Conference 2020*, pages 917–927, Taipei, Taiwan, April 2020. ACM.

[Lin *et al.*, 2013] Wangqun Lin, Xiangnan Kong, Philip S Yu, Quanyuan Wu, Yan Jia, and Chuan Li. Community detection in incomplete information networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 341–350, New York, NY, USA, April 2013. ACM.

[MacMahon and Garlaschelli, 2015] Mel MacMahon and Diego Garlaschelli. Community detection for correlation matrices. *Physical Review X*, 5(2):021006, 2015.

[Newman and Girvan, 2004] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[Newman, 2018] Mark EJ Newman. *Networks*. Oxford university press, 2018.

[Peixoto, 2014] Tiago P Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4(1):011047, 2014.

[Peixoto, 2018] Tiago P. Peixoto. Reconstructing networks with unknown and heterogeneous errors. *Physical Review X*, 8(4):041011, Oct 2018.

[Rozemberczki *et al.*, 2018] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. Gemsec: Graph embedding with self clustering. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 65–72, New York, NY, USA, August 2018. ACM.

[Shao *et al.*, 2019] Junming Shao, Zhong Zhang, Zhongjing Yu, Jun Wang, Yi Zhao, and Qinli Yang. Community detection and link prediction via cluster-driven low-rank matrix completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3382–3388, New York, NY, USA, July 2019. IJCAI.

[Tran *et al.*, 2018] Cong Tran, Wonyong Shin, and Andreas Spitz. Community detection in partially observable social networks. *arXiv: Social and Information Networks*, 2018.

[Wang *et al.*, 2017] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 203–209. AAAI, December 2017.

[Yang and Leskovec, 2013] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 587–596. ACM, February 2013.

[Ye *et al.*, 2018] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1393–1402, USA, October 2018.

[Zhang *et al.*, 2015] Hongyi Zhang, Irwin King, and Michael Lyu. Incorporating implicit link preference into overlapping community detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, USA, January 2015.