Applications on Stochastic Block Model and Markov Chain Monte Carlo based recovery method

Wenxuan Zhang[a]

[a]*School of Art and Science, University of Pennsylvania*

Keyword: Community detection, Stochastic block model, Markov Chain Monte Carlo

Abstract: Detecting the communities in a complex system is a crucial problem in many fields of research. This problem can be modelled and solved by a generative model, stochastic block model. This project introduced some basic concepts of stochastic block model, as well as theorems relating to recovering the group labels from connection information. An improved Markov Chain Monte Carlo method from Peixot[Pei14a] was presented then to solve the recovery problem. At last, the author constructed a model of FBS American football team network, and proved it to be solvable in partial recovery, and recovered the system by MCMC. The numerical results both from the model data and the real world data with noise satisfied the theoretical result.

1. Introduction

The concepts of network are common in many scientific fields including sociology, biology, computer science, and physics. For a complex system, when we viewing the objects as vertices and the connections between object pairs as edges, it is a network. In the field of network research, it is widely accepted that any network tends to have inner structures, one of which is community. Objects in the same community will have tighter connections and as a result they will have similar properties, so that we can study the network by parts or make use of the commonality inside a community. Community detection can be useful in many real world networks, such as online social works, recommend system, and all of physical and life sciences.

Stochastic block model(SBM) is a canonical and popular generative model for community detection tasks. From the mathematical point-view, community detection requires clustering vertices in a graph. Under this context, stochastic block model can be described as follows, the nodes in a graph are labeled from some probability law, once the vertices get their labels, connections will appear in some probability depending on the labels of both end vertices. In a sense, it simulates the generation of a network–the frequency of communication between objects depending on their group membership. Obviously, the stochastic block model represents a good fit for the real data, and it also can be extended to a advanced refinement. However, what we see mostly is only the result of the network generation, without the knowledge of vertex labels, and such membership are sometimes the goal of study. Fortunately, algorithms are provided to recover the labels of a network by observing the connectivity. And scientists also proved that such recovery are almost correct in some standards.

This project firstly introduced recent developments of algorithms to recovery the model in the section 2. Then, the precise introduction of stochastic block model and Markov chain Monte Carlo algorithm was presented in section 3,4. And in section 5, the author used the theories and algorithms above to evaluate the network of American FBS football teams and recovered the communities from their connectivity.

2. Literature Review

In the last few years, algorithms for classical stochastic block model based on matrix computation were widely used. In 2012, Raj and M.E.J[NN12] analyzed the spectral properties of the adjacency and modular matrices and demonstrated that spectral modular maximization is an optimal detection method in recovering the communities in stochastic block model. And then in 2013, M.E.J[New13] showed that two of the most popular methods, likelihood maximization and Spectral algorithm, can be mapped directly onto versions of the standard minimum-cut graph partitioning problem, so that we can use any of the many well-understood partitioning algorithms. In this context, many other researchers[YP14, CRV15] applied this method and proposed robust and accurate refinements.

Other heuristic algorithms were also proved to be accurate and efficient in identifying the community, such as algorithms based on random walk[FMP19]

Recently, algorithms based on statistical inference made a great process. Recent advances in stochastic gradient Markov Chain Monte Carlo(MCMC) have played a crucial role in improving the stability of these techniques. As a result, many improved algorithms[PC19, LS19] based MCMC sprang up.

Aside from classical stochastic block model, in some disciplines, many large graphs are believed to have hierarchical structures, so the algorithms[LTA$^+$15, PAL19] for hierarchical stochastic block models made good performance for these graphs.

3. Stochastic Block Models

In this section, we introduced basic concepts and theorems about stochastic block model. Most contents are from Abbe's work[Abb17].

3.1. The General Stochastic Block Model

Definition 1. (Stochastic block model)*Let n be a positive integer (the number of vertices), k be a positive integer (the number of communities), $\mathbf{p} = (p_1, \dots, p_k)$ be a probability vector on $[k] := 1, \dots, k$ (the label of the k communities) and $\mathbf{W}$ be a $k \times k$ symmetric matrix with entries in $[0, 1]$ (the connectivity probabilities). The pair $(\mathbf{X}, G)$ is drawn under $SBM(n, \mathbf{p}, \mathbf{W})$ if $\mathbf{X}$ is an n-dimensional random vector with i.i.d. components distributed under $\mathbf{p}$, and G is an n-vertex simple graph where vertices i and j are connected with probability $W_{\mathbf{X}_i,\mathbf{X}_j}$, independently of other pairs of vertices. We also define the community sets by $n_i = n_i(\mathbf{X}) := \{v \in [n] : X_v = i\}, i \in [k]$.*

Here note that $\mathbf{W}$ is not the adjacent matrix of the graph, and $\mathbf{W}_{ij}$ denotes the probability of connection between community i and community j

The idea of the model is to generate edges according to vertices labels. More precisely, when generating a graph with n vertices, vertex i gain the label $x_i = l$ with probability p_l. Once all the vertices gain their labels $\mathbf{x}$, the connections between a pair of vertices i, j appears following the Bernoulli law with mean W_{x_i,x_j}, depending the label of i, j. Thus the distribution of $(\mathbf{X}, G)$, where $G = ([n], E(G))$ is defined as follows, for label $\mathbf{x} \in [k]^n$ and connections$\mathbf{y} \in \{0, 1\}^{\binom{n}{2}}$

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) := \prod_n p_{x_n} \tag{1}$$

$$\mathbb{P}(E(G) = \mathbf{y}|\mathbf{X} = \mathbf{x}) := \prod_{1 \le i < j \le n} W_{x_i,x_j}^{y_{ij}} (1 - W_{x_i,x_j})^{(1-y_{ij})} \tag{2}$$

Here $y_{ij} = 1$ if there exists an edge between x_i and x_j, and $y_{ij} = 0$ otherwise.

Once we know the labels, we get the information of edge counts $\{e_{st}\}$ and node counts $\{n_k\}$of every group. Then we can factor the same part in Equation (1, 2) and get

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \prod_k p_k^{n_k} \tag{3}$$

$$\mathbb{P}(E(G) = \mathbf{y}|\mathbf{X} = \mathbf{x}) = \prod_{1 \le s < t \le k} W_{x_s,x_t}^{e_{st}} (1 - W_{x_s,x_t})^{e_{st}^c} \tag{4}$$

where e_{st} is the edge count between groupss, t, e_{st}^c is the edges that didn't generate. Formally,

$$\begin{aligned} e_{st}^c &= \{\text{Maximum possible edges between s and t}\} - e_{st} \\ &= 1_{\{s=t\}}(\frac{n_s(n_s - 1)}{2} - e_{st}) + 1_{\{s \neq t\}}(n_s n_t - e_{st}) \end{aligned}$$

In the real world, some systems have symmetric properties, that is, vertices have the same probability to be in any communities, and the probability of the connections between two vertices only depends on if they are in the same community. More precisely, $\mathbf{W}$ takes the same value on the diagonal and the same value outside the diagonal for such system. Here we introduced the symmetric stochastic block model(SSBM).

Definition 2. (Symmetric stochastic block model)$(\mathbf{X}, G)$ *is drawn under $SSBM(n, k, A, B)$, if $\mathbf{p} = \{\frac{1}{k}\}^k$ and $\mathbf{W}$ takes value A on the diagonal and B off the diagonal.*

3.2. Recovery and Convergence

In the real world, under most circumstances we will see the connectivity of the objects. If we assume the system is generated following the stochastic block model, then the goal of community detection is to recover the labels $\mathbf{X}$ by observing the connections, that is the graph$G$, up to some level of accuracy. Here the author introduced the formal definition of recovery in different levels of accuracy, and theorems that whether a system can be solved in such levels.

Definition 3. *The* **agreement** *between two community vectors $x, y \in [k]^n$ is obtained by maximizing the common components between x and any relabelling of y, i.e.,*

$$A(x,y) = \max_{\pi \in S_k} \frac{1}{n} \sum_{n} \mathbf{1}(x_i = \pi(y_i))$$

where S_k is the group of permutations on $[k]$.

Definition 4. *Given a probability space on which SBM is defined, and on which a deterministic recovery algorithm is run and takes G as an input and outputs $\hat{X} = \hat{X}(G)$, the following recovery requirements are solved if the probability of $A(X, \hat{X})$ satisfies*

- **Exact recovery**: $\mathbb{P}\{A(X,\hat{X}) = 1\} = 1 - o(1)$,
- **Almost exact recovery**: $\mathbb{P}\{A(X,\hat{X}) = 1 - o(1)\} = 1 - o(1)$,
- **Partial recovery**: $\mathbb{P}\{A(X,\hat{X}) \geq \alpha\} = 1 - o(1), \alpha \in (0,1)$.

In other words, exact recovery requires the entire partition to be correctly recovered, almost exact recovery allows for a vanishing fraction of errors in partition and partial recovery allows for a constant fraction of errors in partition. We call α the agreement or accuracy of the algorithm.

From the definitions of different recovery levels and our previous experience, it is obvious that exact recovery $\implies$ almost recovery $\implies$ partial recovery.

Next the author introduced theorems that decide whether a graph can be recovered in some levels of accuracy.

Theorem 1. *Exact recovery in $SBM(n, p, \frac{\ln(n)}{n}Q)$ is solvable and efficiently so if*

$$I^+(p,Q) := \min_{1 \leq i < j \leq n} D_+((diag(p)Q)_i || (diag(p)Q)_j) > 1, \quad 1 \leq i < j \leq k$$

and is not solvable if $I^+(p,Q) < 1$, where D_+ is defined by

$$D_+(A||B) := \max_{t \in [0,1]} D_t = \max_{t \in [0,1]} \sum_x A(x) f_t(\frac{A(x)}{B(x)}), \quad f_t(y) := 1 - t + ty - y^t$$

Note that in the symmetric case $SSBM(n, k, a\frac{\ln n}{n}, b\frac{\ln n}{n})$, the D_t is maximized at the value of $t = 1/2$, and it reduces in this case to the Hellinger divergence between any two columns of Q, the theorems inequality becomes

$$\frac{1}{k}(\sqrt{a} - \sqrt{b})^2 > 1$$

Almost exact recovery and partial recovery are studies less than exact recovery, and those related theorems for general SBM are not that simple and strong, often some extra conditions are needed. But in the symmetric case, necessary and sufficient conditions have been identified.

Theorem 2. *Almost exact recovery is solvable in $SSBM(n, k, \frac{a_n}{n}, \frac{b_n}{n})$ if and only if*

$$\frac{(a_n - b_n)^2}{k(a_n + (k-1)b_n)} = \omega(1)$$

Here $x_n = \omega(1)$ meas x_n is diverging

Theorem 3. *Partial recovery in $SSBM(n, k, \frac{a}{n}, \frac{b}{n})$ takes place when the following equation is finite:*

$$\frac{(a-b)^2}{k(a + (k-1)b)} = O(1)$$

4. Algorithm: Improved Markov Chain Monte Carlo method

In this section, we introduced a recovery algorithm provided by Peixoto[Pei14a] based on Markov Chain Monte Carlo.

Maximum likelihood

Recovering the model equals to inferring the most likely labels $\mathbf{x}$ among all possible labels $\mathbf{x}'$ given connectivity information $G = ([n], E(G) = \mathbf{y})$, i.e.maximizing the possibility $\mathbb{P}(\mathbf{x}|E(G) = \mathbf{y})$. By Bayesian inference and learning, we have

$$\mathbb{P}(\mathbf{x}|E(G) = \mathbf{y}) = \frac{\mathbb{P}(E(G) = \mathbf{y}|\mathbf{x})\mathbb{P}(\mathbf{x})}{\sum_{\mathbf{x}'} \mathbb{P}(E(G) = \mathbf{y}|\mathbf{x}')\mathbb{P}(\mathbf{x}')} \tag{5}$$

Maximize Equation (5) equal to maximize the numerator $\mathbb{P}(E(G) = \mathbf{y}|\mathbf{x})\mathbb{P}(\mathbf{x}) = \mathbb{P}(E(G) = \mathbf{y}, \mathbf{X} = \mathbf{x})$. By Equation (3,4), each graph with the same edge counts $\{e_{st}\}$ and note counts $\{n_k\}$ is equally likely. We have

$$\begin{aligned}
&\mathbb{P}(E(G) = \mathbf{y}, \mathbf{X} = \mathbf{x}) \\
&= \mathbb{P}(E(G) = \mathbf{y}, \mathbf{X} = \mathbf{x}, \{n_k\}, \{e_{st}\}) \\
&= \mathbb{P}(E(G), \{n_k\}, \{e_{st}\})\mathbb{P}(E(G) = \mathbf{y}, \mathbf{X} = \mathbf{x}|E(G), \{n_k\}, \{e_{st}\}) \\
=&\frac{1}{N_G(\{n_k\}, \{e_{st}\})}\mathbb{P}(E(G), \{n_k\}, \{e_{st}\})
\end{aligned}$$

where $N_G(\{n_k\}, \{e_{st}\})$ is the number of graphs that has the same $\{n_k\}, \{e_{st}\}$ as G. Hence, maximizing the likelihood is identical to computing the micro-canonical entropy

$$S(\{n_k\}, \{e_{st}\}) = \ln N_G(\{n_k\}, \{e_{st}\})$$

which can be converted to

$$S(\{e_{ij}\}, \{n_k\}) = \frac{1}{2} \sum_{1 \le r,s \le k} n_r n_s H_b(\frac{e_{rs}}{n_r n_s})$$

where k is the number of blocks, e_{ij} is the number of edges connecting the nodes in block i and block j, n_r is the number of nodes in block r, $H_b(x) = -x \ln x - (1-x)\ln(1-x)$ is the binary entropy function.

Then the maximum likelihood can be computed by computing $S(\{e_{ij}\}, \{n_k\})$ for all the possible partitions.

However, testing all the partitions can only be achieved in small networks. Instead, Markov Chain Monte Carlo can be used to sample partitions from a probability as a monotone decrease function of $S(\{e_{ij}\}, n)$, and smallest $S(\{e_{ij}\}, n)$ can be obtained by choose the sampled partition with highest frequency. Details are shown in the following part.

4.1. Markov Chain Monte Carlo (MCMC) method

The probability distribution proportional to $e^{-S(\mathbf{x})}$ mentioned above can be sampled by doing the following algorithm, known as MCMC.

In statistics, Markov chain Monte Carlo (MCMC) methods comprise a class of algorithms for sampling from a probability distribution. Generally, in each step, a point will be sampled and added to the Markov Chain with some acceptance rate, that is the part of Monte Carlo. The sampling process depends only on the previous point, that is the part of Markov Chain. When such Markov Chain converges to its equilibrium state, one can obtain a sample of the desired distribution by recording states from the chain.

Random walk Monte Carlo methods and Metropolis - Hastings were used in Peixoto's work. In each step he attempts to move a vertex from block r to s to get a new partition with a probability in a sense $p(r \to s|t)$,

$$p(r \to s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon k}$$

where e_t is the number of edges connecting the block t. This means if a vertex A has a neighbor B in block t, A will get a new label s from B's neighbor, and the probability of getting s is proportion to e_{ts}. See the example in Figure 1 Intuitively, this method attempts to guess the label of a given node by inspecting the label of its neighbors and by using the currently inferred model parameters to choose the most likely blocks to which the original node belongs.

However, for some ϵ, the $p(r \to s|t)$ will not fulfill the detailed balance, but under proper acceptance rate a deriving from $p(r \to s|t)$, the chain can converge to a equilibrium state, and after a sufficiently long

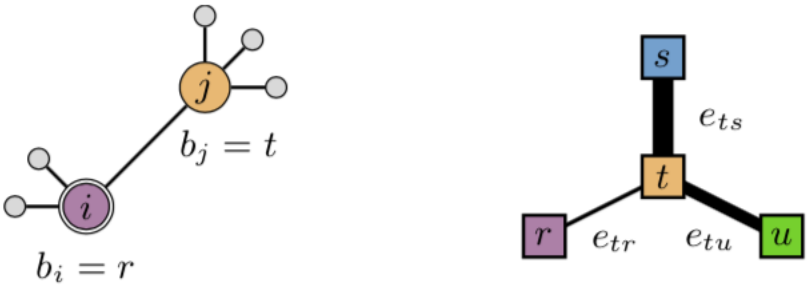


Figure 1: b_i is more likely to be relabeled as s since e_{ts} has the maximum value of b_j's neighborhood

equilibrium time, each observed partition must occur with the desired probability proportional to $e^{-S(\{e_{ij}\},n)}$. The acceptance rate a is

$$a = \min\{e^{-\beta \Delta S_t} \frac{\sum_t p_t^i p(s \to r|t)}{\sum_t p_t^i p(r \to r|t)}, 1\}$$

where p_t^i is the fraction of neighbors of node i which belong to block t, and $p(s \to r|t)$is computed after the proposed $r \to s$ move, whereas $p(r \to s|t)$ is computed before. The parameter β is an chosen inverse temperature.

In order to sample a partition, the process is as follows:

- A random neighbor j of the node i being moved is selected, and its block membership $t = b_j$ is obtained;
- The value s is randomly selected from all k choices with equal probability;
- With probability a it is accepted;
- If it is rejected, a randomly chosen edge adjacent to block t is selected, and the block label s is taken from its opposite endpoint.

4.2. Improvement

Although the such moves above provide a considerable improvement over the fully random alternative whenever the number of blocks k becomes large, there remains an important problem when applying it. Namely the mixing time may be heavily dependent on how close between the start point and equilibrium state. Chances are that the algorithm takes a long time to escape meta-stable states. Here meta-stability means the graph remains in a state where partially recovered for a long time. Before escaping from this state, it is difficult to tell the meta-stable state and the equilibrium state apart. This problem is exacerbated if the average block size n/k increases.

To avoid meta-stable states when block sizes are large, Peixoto found best configuration for block number $k' > k$, then obtain k blocks by merging blocks together from k' blocks. It can be implemented by viewing the blocks as nodes to forming a new graph, where the edges are the edge multiplicities between each block node. In such graph, block merging is just similar as the initially node moves, that is to give the label of selected block(node in the new graph) to the block to be moved.

In order to select the best merges, we attempt n_m moves for each block node, and collectively rank the best moves for all nodes according to $\Delta S(\mathbf{x})$. From this global ranking, we select the best $k' \to k$ merges to obtain the desired partition into k blocks.

If the value of n/k' itself is too large, the same problem raised as before. Therefore we iterate the process by starting with $k_1 = n$, and selecting $k_{i+1} = k_i/\sigma$, until we reach the desired k value, where $\sigma > 1$ controls how greedily the merges are performed.

To diminish the effect of bad merges done in the earlier steps, we also allow individual node moves between each merge step.

5. Applications

In this section, the author applied previous models and algorithms to the network of American football games between Division IA colleges, including constructing and evaluating the theoretical model from the system, as well as using algorithms to recover the system.

5.1. Model construction

The NCAA Division I Football Bowl Subdivision (FBS), formerly known as Division I-A, is the top level of college football in the United States. In regualre season most FBS teams play twelve games, with eight or nine of those games coming against conference opponents. For non-conference regular season games, FBS teams are free to schedule match-ups against any other FBS team, regardless of conference. A small number of FBS teams are independent, and have total control over their own schedule.

Regardless the independent teams, every conference have nearly the same numbers of team members. Then every teams have nearly the same probability to play a game with other teams in their conferences, also with outside their conferences. Then conference system between FBS teams can be viewed as a symmetric stochastic block model with $n = 107$ teams, $k = 11$ conferences, and the parameter in definition 2 can be approximate as

$$A = \frac{8k}{12n}, B = \frac{4k}{12n(k-1)}$$

Do the computation theorem 1, we have $a \approx 1.569, b \approx 0.078$, and

$$(\sqrt{a} - \sqrt{b})^2 \approx 0.946 < k$$

so the exact recovery of this system can not be solvable.

Do the computation in theorem 2, we have

$$\frac{(a_n - b_n)^2}{k(a_n + (k-1)b_n)} = (\frac{3}{4} - \frac{1}{4(k-1)})^2$$

which will not diverge. So this system cannot be solved in almost exact recovery.

Do the computation in theorem 3, we have the value approximate at 0.526, so the system can be solved in partial recovery.

5.2. Numerical Results

In this part the author analized the data-set and use the algorithm above to recover the partition of the system. Data is from Newman[GN02] and revised by Evans[Eva12], which records the football games between FBS teams during regular season Fall 2000. The codes of the algorithm above are provided by Peixoto[Pei14b]

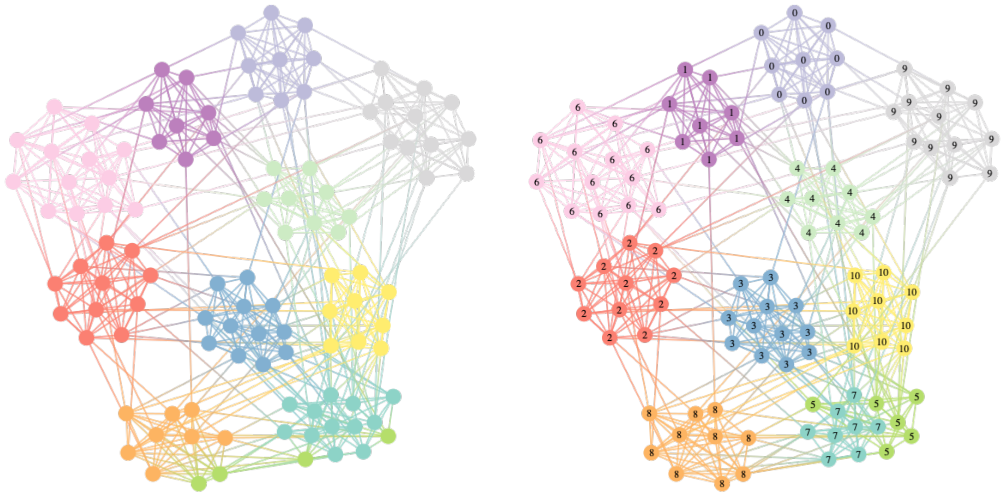
Recovery and Results

The original data include information for independent teams which violates the symmetric assumption, the author got rid of these nodes and edges to fit the model. The partition results and the ground-truth partition is in Figure 2

The author used Normalized Mutual Information (NMI) to evaluate the results. It scales Mutual Information (MI) score between 0 (no mutual information) and 1 (perfect correlation). The NMI score of the algorithm with 100 runs are in Figure 4a. From the figure we can conclude that this algorithm can only achieve partial recovery for the system.

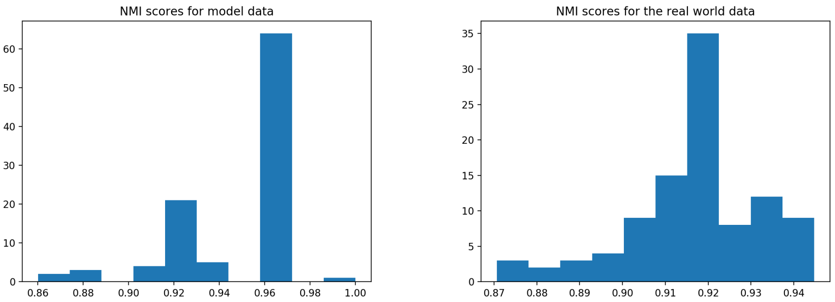
Comparison with the real world data

The original data-set also includes independent teams and their connection with other FBS teams. These information can be viewed as the noisy data to the model. Nevertheless, the algorithm used is robust regarding the noisy. Figure shows the partition result and the ground truth value. Figure4b shows the comparison of the partition and NMI scores between the real world data and data with out noise.

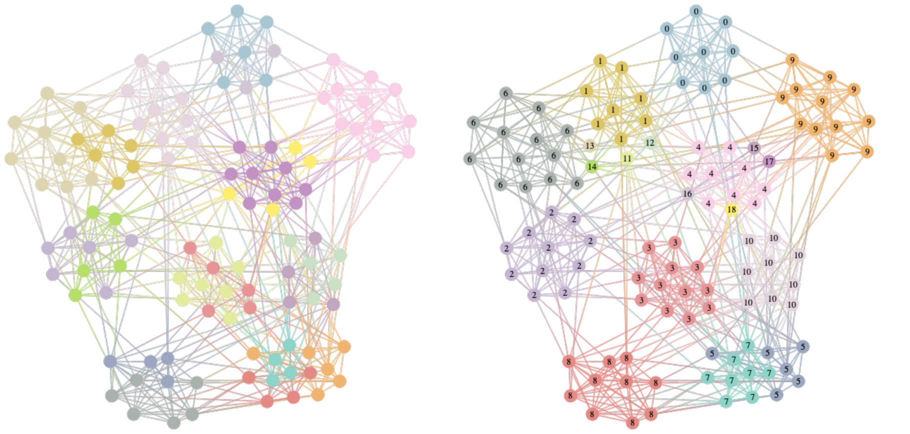


(a) Recovery Result (b) Ground Truth Partition

Figure 2: Partition of model result and the ground truth value



(a) Histogram of NMI for model data in 100 runs (b) Histogram of NMI for the real world data in 100 runs



(a) Recovery Result (b) Ground Truth Partition

Figure 4: Caption

References

[Abb17] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *J. Mach. Learn. Res.*, 18:177:1–177:86, 2017.

[CRV15] Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in the sparse graphs: A spectral algorithm with optimal rate of recovery. *CoRR*, abs/1501.05021, 2015.

[Eva12] Tim Evans. American college football network files, Dec 2012.

[FMP19] Reza Fathi, Anisur Rahaman Molla, and Gopal Pandurangan. Efficient distributed community detection in the stochastic block model. *CoRR*, abs/1904.07494, 2019.

[GN02] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[LS19] Xiaoyan Lu and Boleslaw K. Szymanski. Regularized stochastic block model for robust community detection in complex networks. *CoRR*, abs/1903.11751, 2019.

[LTA$^+$15] Vince Lyzinski, Minh Tang, Avanti Athreya, Youngser Park, and Carey E. Priebe. Community detection and classification in hierarchical stochastic blockmodels. *IEEE Transactions on Network Science and Engineering*, 4:13–26, 2015.

[New13] M. E. J. Newman. Community detection and graph partitioning. *CoRR*, abs/1305.4974, 2013.

[NN12] Raj Rao Nadakuditi and M. E. J. Newman. Graph spectra and the detectability of community structure in networks. *CoRR*, abs/1205.1813, 2012.

[PAL19] Marina S. Paez, Arash A. Amini, and Lizhen Lin. Hierarchical stochastic block model for community detection in multiplex networks. *CoRR*, abs/1904.05330, 2019.

[PC19] S. Pal and M. Coates. Scalable mcmc in degree corrected stochastic block model. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5461–5465, May 2019.

[Pei14a] Tiago P. Peixoto. Efficient monte carlo and greedy heuristic for the inference of stochastic block models. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 89 1:012804, 2014.

[Pei14b] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014.

[YP14] Se-Young Yun and Alexandre Proutière. Accurate community detection in the stochastic block model via spectral algorithms. *CoRR*, abs/1412.7335, 2014.