A Survey of Community Detection in the Stochastic Block Model

Jasjeet Dhaliwal
UMass Amherst
jdhaliwal@cs.umass.edu

Abstract

In this survey, we first describe a generative random graph model called the Stochastic Block Model. We then describe three algorithms that solve the community detection problem in such graphs. First, we present a Bayesian inference method applied by Decelle et al. [1] and then we present two spectral methods created by Ng et al. [2] and McSherry [3] respectively. We contain the survey to illustrate the core concepts behind each algorithm and sketch proof outlines only when required.

1 INTRODUCTION

Online communities, food webs, metabolic networks and genetic regulatory networks form graphs with connections between communities. Some graphs such as social networks display assortative structure, i.e. dense connections between members of the same community and sparse connections between communities. In general, real world networks are not random graphs as they contain non-random structure and the distribution of the vertex degree is usually quite broad [11]. Further, the distribution of edges also displays non-random structure as we find a high concentration of edges between a group of vertices and low number of edges between groups. This property of real world networks is referred to as community structure. Despite the differences between real world networks and random graphs, we still get very good results in inferring properties of the real world networks using random graphs as approximations.

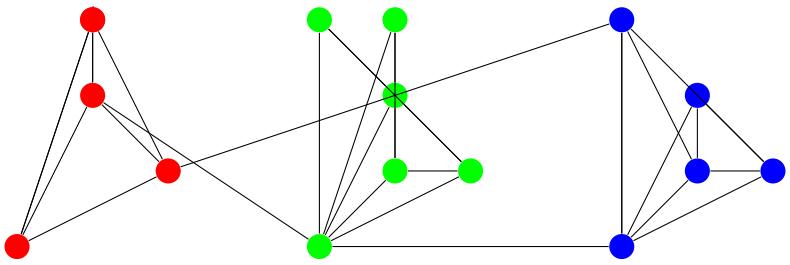

Figure 1: Network with 3 clusters

2 Stochastic Block Model

The Stochastic block model is a random generative model for such networks. The analysis of the model allows us to infer detection thresholds for communities and determine whether a graph's parameters can be inferred, based on the amount of information contained in the graph.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

Stochastic Block Model Let G be a graph with N nodes. Let, each node have a hidden label $t_i = \{1, \ldots, q\}$, specifying group memberships. Each label is chosen independently where n_a is the probability that a given node has a label a in $\{1, \ldots, q\}$, where $\sum_a n_a = 1$. If N_a is the number of nodes in group a, then $n_a = \lim_{N\to\infty} \frac{N_a}{N}$.

After nodes have been assigned labels, the graph is generated as follows:
For all i, j pairs with $i < j$, an edge is added independently between i, j with probability p_{t_i,t_j} and left out with probability $1 - p_{t_i,t_j}$. The resulting matrix of these probabilites is called affinity matrix P_{ab}, where $p_{ab} = \mathcal{O}(\frac{1}{N})$. We now discuss the algorithms used to infer cluster memberships in the stochastic block model.

3 Bayesian Inference: Decelle et al.

Use C_{ab} as the affinity matrix where C_{ab} is a scaled version of P_{ab} such that $c_{ab} = Np_{ab} = \mathcal{O}(1)$.

The input to an algorithm here would be the Adjacency Matrix A_{ij} of the graph and it will output the parameters of the model $\theta = \{q, \{n_a\}, \{p_{ab}\}\}$ and the true group assignments t_i for all nodes in the graph. From Bayesian inference we have,

$P(\theta|G) = \frac{P(\theta)}{P(G)} \sum_{\{t_i\}} P(G, \{t_i\}|\theta)$

Since we do not have any information about the parameters, we assume that $P(\theta)$ is uniform. Therefore, maximizing $P(\theta|G)$ is equivalent to maximizing the sum $\sum_{t_i} P(G, \{t_i\}|\theta)$.

The function $P(G, \{t_i\}|\theta)$ is the likelihood function, i.e. the likelihood that the given model parameters would produce the graph that was observed assuming that the parameters are in fact θ.

Then,

$$P(G, \{t_i\}|\theta) = \prod_i n_{t_i} \prod_{i<j} \left[p_{t_i,t_j}^{A_{ij}} (1 - p_{t_i,t_j})^{1-A_{ij}} \right]$$

Using the above P, we see that the hamiltonian of this graph can be written as:
$H(\{t_i\}|\theta) = -\sum_{ij} logn_{t_i} - \sum_{i<j} \left[A_{ij} logc_{t_i,t_j} + (1 - A_{ij}) log(1 - \frac{c_{t_i,t_j}}{N}) \right]$

where we assume a strong interaction between connected nodes and a weak interaction between unconnected nodes. Now we need to find the most likely θ. This is where belief propagation and mean field theory help us.

3.1 Belief Propagation

A belief at node i corresponds to its estimated marginal: $b(x_i) = k\phi(x_i) \prod_{j\in N(i)} m_{ji}(x_i)$ where k is a normalization constant and $m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{ki}(x_i)$
Similarly, $b(x_i, x_j) = k\psi_{ij}(x_i, x_j)\phi(x_i)\phi_j(x_j) \prod_{k\in N(i)\setminus j} m_{ki}(x_i) \prod_{l\in N(j)\setminus i} m_{kj}(x_j)$

3.2 Free Energies and Mean field theory

Let, $b(\{t_i\})$ be the belief distribution and $p(\{t_i\})$ correspond to the true distribution. Then, we know that the Kullback-Liebler Divergence: $D(b(\{t_i\})||p(\{t_i\})) = \sum_{\{t_i\}} b(\{t_i\}) ln \frac{b(\{t_i\})}{p(\{t_i\})}$

We can then use the Boltzman law and let the parameters k and T be 1. Then, $p(\{t_i\}) = \frac{1}{Z} e^{-E(\{t_i\})}$, where $E(\{t_i\})$ is the energy of the system expressed as the hamiltonian above. $E(\{t_i\}) = -\sum_{(} \{t_i, t_j\}) ln\psi_{\{t_i,t_j\}}(t_i, t_j) - \sum_{t_i} ln\phi_i(t_i)$.

Now, $D(b(\{t_i\})||p(\{t_i\})) = \sum_x b(x)E(x) + \sum_x b(x) lnb(x) + lnZ$ where we refer to the first term as the average energy U, and the second term is called the negative entropy S. Then, the Bethe free energy of the system is defined as $G = U - S$. The key insight from the above equation is that the

Kullback-Liebler divergence is minimum when $G = -lnZ$. Therefore we know that G is bounded below and we can search for the set of beliefs that minimize G.

In our case, we choose to propagate beliefs through the graph until they reach a fixed point. We then use these fixed points of belief propagation and substitute them in the free energy equation. We can now minimize the free energy equation in a fashion similar to the EM algorithm.

1. Initialize θ
2. Evaluate G
3. Update θ to the values that minimize G.
4. Repeat 2,3 until convergence.

We then repeat the algorithm with several values of q' such that if q is the true number of groups then G decreases with q' and then stays constant for $q' \geq q$.

4 Spectral Clustering

Spectral clustering is based on the idea that vertices represented by their eigenvectors contain more information about the true community structure. In cases, where the graph is dense, this community structure can be extracted using the spectrum of a the adjacency matrix with a high degree of confidence. Generally, a spectral algorithm derives the eigenvalues and eigenvectors of a matrix A, after which it represents each vertex in the adjaceny matrix using a $k-$dimensional vector based on its entries in the first $k-$eignevectors. It then uses a clustering algorithm to cluster the vertices based on the eigen-representation. The first two eigenvectors play an especially important role. The first eigenvector sorts vertices based on their degree and the second eigenvector sorts the vertices based on the cluster they belong to. Further, the magnitude of the first eigenvalue gives us important information of about the average degree of the graph. The magnitude of the second eigenvalue determines the connected-ness of a graph. We now define two algorithms that use spectral techniques to infer the clustering of random graphs.

4.1 Spectral Clustering : Ng et al.

Here we present the algorithm by Ng et al. [2]. Given, an adjacency matrix A, we can run the following spectral clustering algorithm for group size k.

1. Compute the normalized Laplacian $L_n = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where $D_{ij} = \sum_j A_{ij}$ for $i = j$ and $D_{ij} = 0$ otherwise. Thus, D is a diagnoal matrix. Further, note that $I - L_n$ would reverse the order of the eigenvalues while not changing the eigenvectors.
2. Compute the larges k eigenvectors $x_1, \ldots, x_k$ of L_n.
3. Stack the k eigenvectors columnwise to form the matrix X.
4. Normalize X to form the matrix Y such that $y_{ij} = \frac{x_{ij}}{(\sum_j (x_{ij})^2)^{\frac{1}{2}}}$.
5. Cluster the points y_i i.e. the rows of Y into k clusters using k-means.
6. Assign node A_i to cluster $j \iff Y_i$ was assigned to cluster j.

In order to understand why this algorithm works, we can examine the ideal case: perfect clusters with no inter-cluster edges. We can explain this with a a group size q of 3.

Call the adjacency matrix here as $\hat{A}$, the Laplacian as L, and the matrix of eigenvectors X, then:

$$A = \begin{bmatrix} A^{(11)} & 0 & 0 \\ 0 & A^{(22)} & 0 \\ 0 & 0 & A^{(33)} \end{bmatrix}; L = \begin{bmatrix} L^{(11)} & 0 & 0 \\ 0 & L^{(22)} & 0 \\ 0 & 0 & L^{(33)} \end{bmatrix}; X = \begin{bmatrix} x_1^{(1)} & 0 & 0 \\ 0 & x_1^{(2)} & 0 \\ 0 & 0 & x_1^{(3)} \end{bmatrix}$$

where we use the following convention: $A^{(ii)}$ is the adjacency matrix for cluster i. Similarly, $L^{(ii)}$ is the normalized laplacian for cluster i. $X \in \mathbb{R}^{nx3}$ and x_1^i corresponds to the eigenvector with

eigenvalue 1. Since 1 is a repeated eigenvalue of L, we can get many solutions to the equations $Lx = (1)x$ and thus we can pick any rotation of X i.e. $\hat{X} = XR$ for some orthogonal matrix R which spans the same subspace as X. Thus we can normalize X and determine Y as :

$$Y = \begin{bmatrix} Y^{(1)} \\ Y^{(2)} \\ Y^{(3)} \end{bmatrix} = \begin{bmatrix} \vec{1} & 0 & 0 \\ 0 & \vec{1} & 0 \\ 0 & 0 & \vec{1} \end{bmatrix} R$$

Therefore, $y_j^i = r_i$ for $j = 1, \ldots, n_i$ and $i = 1, \ldots, k$. This implies that each point in Y will cluster around the corresponding point r_i which is orthogonal to the other points $r_{j \neq i}$ and hence give us a clustering of the dataset.

Since we do not get the ideal case in reality, we can take a pertubation of the ideal case and define an error $E = \hat{A} - A$, where $\hat{A}$ is the adjacency matrix given to us. The error E measures how perturbed the adjacency matrix given to us is. We can then use matrix pertubation theory to determine when the above algorithm will give us accurate results. Using results from matrix pertubation theory we look at the eigengap between the k^{th} and $(k+1)^{st}$ eigenvalue and note that if the eigengap is sufficiently large between the k^{th} and the $(k+1)^{st}$ eigenvalue. $\delta = |\lambda_k - \lambda_{k+1}| > c$ for some large constant c, then we will be able to get accurate clustering on the graph.

4.2 Spectral Clustering: McSherry

In order to present McSherry's algorithm clearly, we use the notation as described in sherry to model random graphs.

Define a random graph $G(\psi, P)$ as follows: $\psi : \{1, \ldots, n\} \mapsto \{1, \ldots, k\}$ be a partition of n nodes into k classes. Let P be a k x k matrix where $P_{i,j} \in [0, 1]$ for all i, j. Include, edge (u, v) with probability $P_{\psi(u),\psi(v)}$.

Now, let $\hat{G}$ be the adjacency matrix given to us. We are also given an input parameter τ which acts as a threshold parameter. Then, we can cluster the vertices in k different clusters using the below algorithm:

1. Split the vertices of $\hat{G}$ randomly into two parts. The resulting matrix will be $\hat{G} = [\hat{A}|\hat{B}]$.
2. Define P_A to be the projection matrix onto the first k eigenvectors of $\hat{B}$ and let P_B be the projection matrix onto the first k eigenvectors of $\hat{A}$.
3. Let $\hat{H} = [P_B(\hat{A}) \mid P_A(\hat{B})]$.
4. While there are unpartitioned nodes:
   (a) Choose an unpartitioned node u_i arbitrarily
   (b) For each v, set $\psi(v) = i$ if $||\hat{H}_{u_i} - \hat{H}_v|| \leq \tau$
5. Return the partition ψ.

The key insight that makes the algorithm work is based on the expectation that $||G - \hat{G}||$ will be small. In order to prove the above, we begin by observing that for a given ψ and P, G has rank k. Therefore, if we define a matrix P_G as the projection matrix onto the columns of G, then we have :

$$\begin{aligned} ||P_G(G) - G|| &= 0 \\ ||P_G(G - \hat{G})|| &= ||P_G G - P_G \hat{G}|| \\ &= ||P_G G - P_G G|| \pm \epsilon \\ &= \pm\epsilon \end{aligned}$$

Since we don't have P_G, the goal is now to find a P_X such that :

$$\begin{aligned} 0 \leq &||P_X(G) - G|| \leq \nu \\ &||P_X(G - \hat{G})|| = \pm\epsilon \end{aligned}$$

for some small error terms ν, ϵ. If we are able to do so, then given that for two columns u, v of G, if $||G_u - G_v|| \gg \epsilon$ then we can simply cluster the columns of G in a greedy manner.

Since the algorithm uses the matrices $\hat{A}$ and $\hat{B}$, we simply need to show that $||A_u - P_B\hat{A}_u||$ is small. Here we use the triangle inequality:

$$||A_u - P_B\hat{A}_u|| \leq ||A_u - P_BA_u|| + ||P_B(A_u - \hat{A}_u)||$$

A standard result from linear algebra states that for any rank k matrix X, $||M - P_M M|| \leq ||M - X||$ where P_M is the projection matrix onto the first k eigenvectors of M.

We will therefore be choosing the matrices that contain the first k eigenvectors of $\hat{A}$ and $\hat{B}$, lets call them P_A and P_B. In order to validate the use of these as the projection matrices we present a theorem that extends the result above. (without sketching its proof).

Theorem 4.1 *For any matrix $\hat{M}$ and rank k matrix M:*

$$||M - P_{\hat{M}}\hat{M}||_F \leq 8k||M - \hat{M}||^2$$

Next, in order facilitate our discussion clearly, we assume (wrongly) that G is split randomly into equal sized partitions such that each partition contains half of the total number of vertices for each group. Therefore each partition is a rank k matrix and further $A = B$.

Then,

$$\begin{aligned} ||A_u - P_BA_u|| &= ||(I - P_B)\hat{B}_u - (I - P_B)(\hat{B}_u - A_u)|| \\ &= ||\hat{B}_u - P_B\hat{B}_u - (I - P_B)(\hat{B}_u - B_u)|| \\ &\leq 2||\hat{B} - B|| \end{aligned}$$

In order to see how we derive the last line of the above equation, note that : $||\hat{B}_u - P_B\hat{B}_u|| \leq ||\hat{B}_u - M|| = ||\hat{B}_u - B||$ for any rank k matrix M and therefore for B.

For the second term we first see that $I - P_B$ is also a projection matrix with the kernel and column space of P_B reversed. Next, we use a standard result from linear algebra which states for any orthogonal projection matrix P : $||Px|| \leq ||x||$. Using these two facts, we can infer the last equation above.

We now need to show that $||\hat{B} - B||$ is also small. In order to do so, McSherry presents a well known expectation result of Furedi and Komlos [12], combined with a recent concentration result of Krivelevich and Vu [13].

Theorem 4.2 *Let $\hat{M}$ be a matrix generated by randomly rounding the entries of a matrix of probabilities M preserving symmetry. Let σ be the largest deviation of an entry in $\hat{M}$. If $\sigma \gg \frac{log^6 n}{n}$ then,*

$$||M - \hat{M}|| \leq 4\sigma\sqrt{n}$$

with probability at least $1 - 2e^{\sigma^2 \frac{2n}{8}}$

Therefore, $||\hat{B} - B||$ will be small as long as our upper bound on the deviation for an entry for G holds true. Next, we prove that $||P_B(A_u - \hat{A}_u)||$ is also small.

Theorem 4.3 (Azuma's Inequality) *Let $\{X_i\}$ be zero mean independent random variables such that $|X_i| \leq c_i$ and $\sum_i c_i^2 \leq 1$. Then, for any $\lambda > 0$ we have ,*

$$P\left[|\sum_i X_i| \geq \lambda\right] \leq 2e^{\frac{-\lambda^2}{2}}$$

In order to prove $||P_B(A_u - \hat{A}_u)||$ is bounded, we use Azuma's inequality along with a union bound. First assume we have a set of orthonormal eigenvectors $\{Q_i\}$. Then:

$$||P_B(A_u - \hat{A}_u)||^2 = \sum_i < (A_u - \hat{A}_u), Q_i >^2$$

$$< (A_u - \hat{A}_u), Q_j > = \sum_v (A_{vu} - \hat{A}_{vu}), Q_{vi})$$

Where each term in the sum of the RHS in the last equation is a zero mean independent random variable. Further $\sum_v (A_{vu} - \hat{A}_{vu}), Q_{vi})^2 \leq 1$ since Q_i is a unit vector. Thus, we can use Azuma's inequality to bound $||P_B(A_u - \hat{A}_u)||$ as well.

5 Implementation

We implemented the spectral clustering algorithm of Ng et al. on the research collaboration graph from UMass Amherst shown in Fig. 2. After trying different values of k, we noted that only for $k = 5$ does the algorithm provide apprixately equal sized clusters. We present the result of the implementation as a graph in Fig. 3.

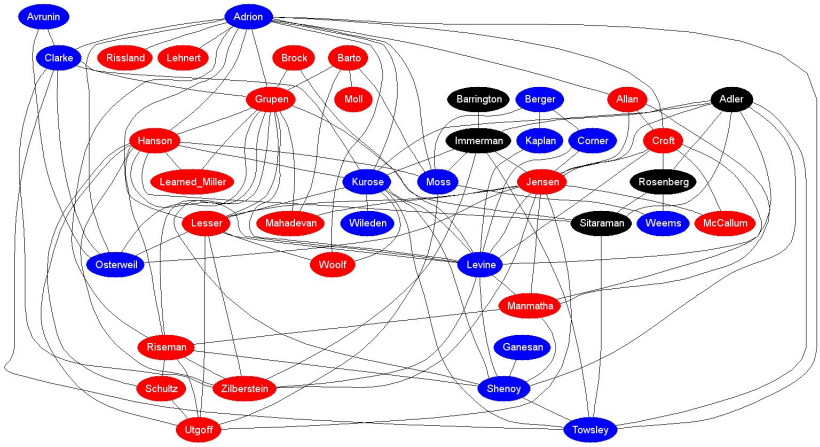

Figure 2: Research collaboration Graph UMass Amherst

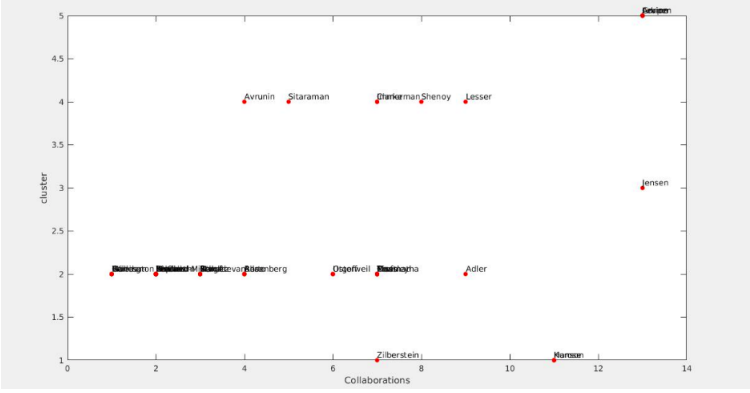

Figure 3: Spectral Clustering with k = 5

6 References

[1] A. Decella, F. Krakala,C. Moore, V. Zbedoborov. Phase transition in the detection of modules in sparse networks. Phys. Rev. Lett. 107, 065701 (2011).

[2] A. Ng, M. Jordan, Y. Weiss. On spectral clustering: Analysis and an algorithm. NIPS, 2002

[3] F. McSherry. Spectral partitioning of random graphs, IEEE Symposium on Foundations of Computer Science, 2001.

[4] M. Hastings, Community Detection as an Inference Problem. Phys. Rev. E 74, 035102 (2006).

[5] J. Yedidia, W. Freeman, Y. Weiss. Understanding Belief Propogations and its Generalizations. In Exploring artificial intelligence in the new millennium pp. 239 - 269

[6] J. Yedidia, W. Freeman, Y. Weiss. Understanding Belief Propogations and its Generalizations. In Exploring artificial intelligence in the new millennium pp. 239-269

[7] A. Coja-Oghlan. Graph Partitioning via Adaptive Spectral Techniques. Journal of Combinatorics, Probability and Computing. Vol. 19(2): pp. 227-284.

[8] F. Krzakalaa,C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, P. Zhang. Spectral redemption: clustering sparse networks. Proceedings of the National Academy of Sciences 110, no. 52 (2013): 20935-20940

[9] A. Decella, F. Krakala,C. Moore, V. Zbedoborov. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys. Rev. E 84, 066106 (2011)

[10] A. Ghasemian, P. Zhang, A. Clauset, C. Moore, L. Peel. Detectability thresholds and optimal algorithms for community structure in dynamic networks. Phys. Rev. X6, 031005 (2016)

[11] S. Fortunato. Community detection in graphs. Physics Reports 486, 75-174 (2010)

[12] F. Zoltanuredi, J. Komlos. The eigenvalues of random symmetric matrices, Combinatorica 1 (1981), no. 3, 233–241.

[13] M. Krivelevich, V. Vu. On the concentration of eigenvalues of random symmetric matrices. Microsoft Technical Report, no. 60, 2000.

[14] E. Abbe, A. Bandeira, G. Hall. Exact Recovery in the Stochastic Block Model. arXiv:1405.3267

[15] E. Abbe, C. Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. arXiv:1512.09080