Community Detection Methods for Generalized Random Dot Product Graphs

Dissertation Proposal

John Koo[1] Michael Trosset[1] Minh Tang[2] Julia Fukuyama[1] Roni Khardon[3] Fangzheng Xie[1]

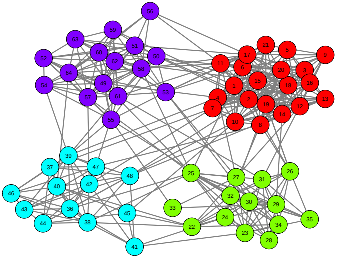
[1]Indiana University, Department of Statistics

[2]North Carolina State University, Department of Statistics

[3]Indiana University, Department of Computer Science

June 2021

Community Detection for Networks



How can we cluster the nodes of a network?

Statistical inference (parametric approach):

1. Define a generative model for graph $G \mid z_1, ..., z_n, \vec{\theta} \sim P(\vec{z}, \vec{\theta})$.
2. Develop a method for obtaining estimators $f(G) = \hat{z}_1, ..., \hat{z}_n$.
3. Identify asymptotic properties of estimators.

Overview

1. Probability Models for Networks
    a. Block Models and Community Structure
    b. (Generalized) Random Dot Product Graphs
    c. Connecting Block Models to the (G)RDPG
2. Popularity Adjusted Block Model
    a. Connecting the PABM to the GRDPG
    b. Orthogonal Spectral Clustering
    c. Sparse Subspace Clustering
3. Community Detection for the (G)RDPG
    a. Manifold Clustering
    b. Manifolds as (G)RDPG Latent Configurations

Probability Models for Networks

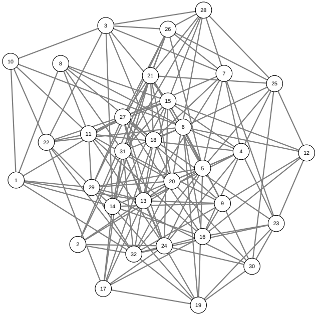
Bernoulli Graphs

Let $G = (V, E)$ be an undirected and unweighted graph with $|V| = n$.
G is described by adjacency matrix A such that $A_{ij} = \begin{cases} 1 & \exists \text{ edge between } i \text{ and } j \\ 0 & \text{else} \end{cases}$
$A_{ji} = A_{ij}$ and $A_{ii} = 0 \ \forall i, j \in [n]$.

$A \sim \text{BernoulliGraph}(P)$ iff:

1. $P \in [0,1]^{n \times n}$ describes edge probabilities between pairs of vertices.
2. $A_{ij} \overset{\text{ind}}{\sim} \text{Bernoulli}(P_{ij})$ for each $i < j$.

Example 1: If every entry $P_{ij} = \theta$, then $A \sim \text{BernoulliGraph}(P)$ is an Erdos-Renyi graph. For this model, $A_{ij} \overset{\text{iid}}{\sim} \text{Bernoulli}(\theta)$.

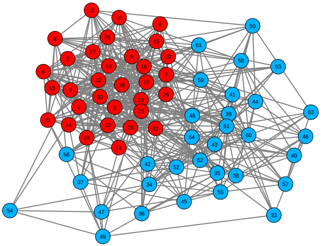


Block Models

Suppose each vertex $v_1, ..., v_n$ has hidden labels $z_1, ..., z_n \in [K]$, and each P_{ij} depends on labels z_i and z_j.
Then $A \sim \text{BernoulliGraph}(P)$ is a *block model*.

Example 2: Stochastic Block Model with two communities

- $z_1, ..., z_n \in \{1, 2\}$
- $P_{ij} = \begin{cases} p & z_i = z_j = 1 \\ q & z_i = z_j = 2 \\ r & z_i \neq z_j \end{cases}$
- To make this an assortative SBM, set $pq > r^2$.
- In this example, $p = 1/2$, $q = 1/4$, and $r = 1/8$.



Block Models

Erdos-Renyi Model (1959)

- $P_{ij} = \theta$
- Not a block model

Stochastic Block Model (Lorrain and White, 1971)

- $P_{ij} = \theta_{z_i z_j}$
- $K(K+1)/2$ parameters θ_{kl}

Degree Corrected Block Model (Karrer and Newman, 2011)

- $P_{ij} = \theta_{z_i z_j} \omega_i \omega_j$
- $K(K+1)/2 + n$ parameters θ_{kl}, ω_i

Popularity Adjusted Block Model (Sengupta and Chen, 2017)

- $P_{ij} = \lambda_{i z_j} \lambda_{j z_i}$
- Kn parameters λ_{ik}

Hierarchy of Block Models

PABM $\rightarrow$ DCBM: $\lambda_{ik} = \sqrt{\theta_{z_i k}}\omega_i$

DCBM $\rightarrow$ SBM: $\omega_i = 1$

SBM $\rightarrow$ Erdos-Renyi: $\theta_{kl} = \theta$

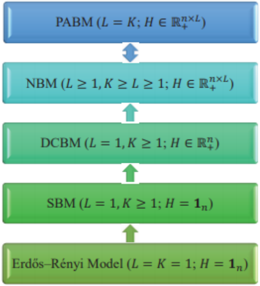


Figure 2: The hierarchy of block models

Majid Noroozi and Marianna Pensky, 2021

(Generalized) Random Dot Product Graph Model

Random Dot Product Graph $A \sim \mathsf{RDPG}(X)$
(Young and Scheinerman, 2007)

- Latent vectors $x_1, ..., x_n \in \mathbb{R}^d$ such that $x_i^\top x_j \in [0,1]$
- $A \sim \mathsf{BernoulliGraph}(XX^\top)$ where $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$

Generalized Random Dot Product Graph $A \sim \mathsf{GRDPG}_{p,q}(X)$
(Rubin-Delanchy, Cape, Tang, Priebe, 2020)

- Latent vectors $x_1, ..., x_n \in \mathbb{R}^{p+q}$ such that $x_i^\top I_{p,q} x_j \in [0,1]$ and $I_{p,q} = \mathsf{blockdiag}(I_p, -I_q)$
- $A \sim \mathsf{BernoulliGraph}(X I_{p,q} X^\top)$ where $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$

If latent vectors $X_1, ..., X_n \overset{\text{iid}}{\sim} F$, then we write
$(A, X) \sim \mathsf{RDPG}(F, n)$ or $(A, X) \sim \mathsf{GRDPG}_{p,q}(F, n)$.

(Generalized) Random Dot Product Graph Model

Recovery/Estimation

Want to estimate X from A, or alternatively, interpoint distances, inner products, or angles.

Adjacency Spectral Embedding

To embed in $\mathbb{R}^d$,

1. Compute $A \approx \hat{V}\hat{\Lambda}\hat{V}^\top$ where $\hat{\Lambda} \in \mathbb{R}^{d \times d}$ and $\hat{V} \in \mathbb{R}^{n \times d}$.
   For RDPG, use d greatest eigenvalues;
   for GRDPG, use p most positive and q most negative eigenvalues.
2. For RDPG, let $\hat{X} = \hat{V}\hat{\Lambda}^{1/2}$;
   for GRDPG, let $\hat{X} = \hat{V}|\hat{\Lambda}|^{1/2}$.

RDPG: $\max_i \|\hat{X}_i - W_n X_i\| \overset{a.s.}{\to} 0$ (Athreya et al., 2018)

GRDPG: $\max_i \|\hat{X}_i - Q_n X_i\| \overset{a.s.}{\to} 0$ (Rubin-Delanchy et al., 2020)

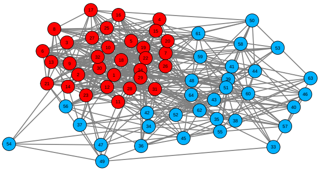
Connecting Block Models to the (G)RDPG Model

All Bernoulli Graphs are RDPG (if P is positive semidefinite) or GRDPG (in general).

Example 2 (cont'd): Assortative SBM ($pq > r^2$) with $K = 2$

$$P_{ij} = \begin{cases} p & z_i = z_j = 1 \\ q & z_i = z_j = 2 \\ r & z_i \neq z_j \end{cases}$$

$$P = \begin{bmatrix} P^{(11)} & P^{(12)} \\ P^{(21)} & P^{(22)} \end{bmatrix} = XX^\top$$

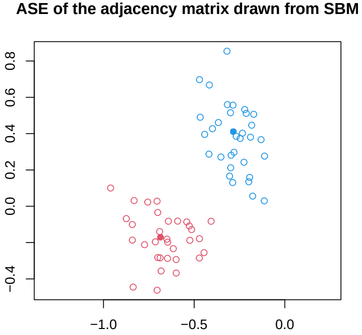


$$X = \begin{bmatrix} \sqrt{p} & 0 \\ \vdots & \vdots \\ \sqrt{p} & 0 \\ \sqrt{r^2/p} & \sqrt{q - r^2/p} \\ \vdots & \vdots \\ \sqrt{r^2/p} & \sqrt{q - r^2/p} \end{bmatrix}$$

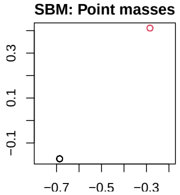
Connecting Block Models to the (G)RDPG Model

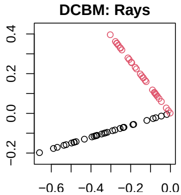
Example 2 (cont'd): If we want to perform community detection,

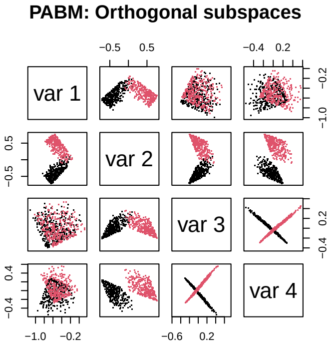
1. Note that $A \sim \text{BernoulliGraph}(P)$ is a RDPG since $P = XX^\top$.
2. Compute the ASE $A \approx \hat{X}\hat{X}^\top$ with $\hat{X} = \hat{V}\hat{\Lambda}^{1/2}$.
3. Apply clustering algorithm (e.g., K-means) to $\hat{X}$, noting that as $n \to \infty$, the ASE approaches point masses.



Connecting Block Models to the (G)RDPG Model







Popularity Adjusted Block Model

Popularity Adjusted Block Model

Def Popularity Adjusted Block Model (Sengupta and Chen, 2017):

Let each vertex $i \in [n]$ have K popularity parameters $\lambda_{i1}, ..., \lambda_{iK} \in [0, 1]$. Then $A \sim \text{PABM}(P)$ if each $P_{ij} = \lambda_{iz_j} \lambda_{jz_i}$, e.g., if $z_i = k$ and $z_j = l$, $P_{ij} = \lambda_{il} \lambda_{jk}$.

Lemma (Noroozi, Rimal, and Pensky, 2020):

A is sampled from a PABM if P can be described as:

1. Let each $P^{(kl)}$ denote the $n_k \times n_l$ matrix of edge probabilities between communities k and l.
2. Organize popularity parameters as vectors $\lambda^{(kl)} \in \mathbb{R}^{n_k}$ such that $\lambda_i^{(kl)} = \lambda_{k_i l}$ is the popularity parameter of the i^{th} vertex of community k towards community l.
3. Each block can be decomposed as $P^{(kl)} = \lambda^{(kl)} (\lambda^{(lk)})^\top$.

Notation: $A \sim \text{PABM}(\{\lambda^{(kl)}\}_K)$.

Connecting the PABM to the GRDPG ($K = 2$)

Theorem (KTT): Given popularity vectors $\{\lambda^{(kl)}\}_2$, we can define block diagonal $X \in \mathbb{R}^{n \times 4}$ and $U \in \mathbb{O}(4)$ such that $A \sim \text{PABM}(\{\lambda^{(kl)}\}_2)$ is equivalent to $A \sim \text{GRDPG}_{3,1}(XU)$.

Proof: Decompose P as follows

$$X = \begin{bmatrix} \lambda^{(11)} & \lambda^{(12)} & 0 & 0 \\ 0 & 0 & \lambda^{(21)} & \lambda^{(22)} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$P = (XU) I_{3,1} (XU)^\top = \begin{bmatrix} \lambda^{(11)} (\lambda^{(11)})^\top & \lambda^{(12)} (\lambda^{(21)})^\top \\ \lambda^{(21)} (\lambda^{(12)})^\top & \lambda^{(22)} (\lambda^{(22)})^\top \end{bmatrix}$$

Connecting the PABM to the GRDPG

Theorem (KTT): $A \sim$ PABM$(\{\lambda^{(kl)}\}_K)$ is equivalent to $A \sim$ GRDPG$_{p,q}(XU)$ with

- $p = K(K+1)/2$, $q = K(K-1)/2$
- $U \in \mathbb{O}(K^2)$
- $X \in \mathbb{R}^{n \times K^2}$ is block diagonal and composed of $\{\lambda^{(kl)}\}_K$ with each block corresponding to a community.

$$X = \begin{bmatrix} \Lambda^{(1)} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \Lambda^{(K)} \end{bmatrix}$$

$$\Lambda^{(k)} = \begin{bmatrix} \lambda^{(k1)} & \cdots & \lambda^{(kK)} \end{bmatrix}$$

Connecting the PABM to the GRDPG

$$X = \begin{bmatrix} \Lambda^{(1)} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \Lambda^{(K)} \end{bmatrix} \qquad U \in \mathbb{O}(K^2)$$

$$A \sim \mathsf{PABM}(\{\lambda^{(kl)}\}_K) \iff A \sim \mathsf{GRDPG}_{p,q}(XU)$$

Remark 1 (orthogonality of subspaces): If $y_i^\top$ and $y_j^\top$ are two rows of XU corresponding to different communities, then $y_i^\top y_j = 0$.

Remark 2 (non-uniqueness of the latent configuration): If $A \sim \mathsf{GRDPG}_{p,q}(Y)$, then $A \sim \mathsf{GRDPG}_{p,q}(YQ)$ for any Q in the indefinite orthogonal group with signature p, q.

Remark 3: Communities correspond to subspaces even with linear transformation $Q \in \mathbb{O}(p, q)$, but this may break the orthogonality property.

Orthogonal Spectral Clustering

Theorem (KTT): If $P = V\Lambda V^\top$ and $B = nVV^\top$, then $B_{ij} = 0$ if $z_i \neq z_j$.

Algorithm: Orthogonal Spectral Clustering:

1. Let V be the eigenvectors of A corresponding to the $K(K+1)/2$ most positive and $K(K-1)/2$ most negative eigenvalues.
2. Compute $B = |nVV^\top|$ applying $|\cdot|$ entry-wise.
3. Construct graph G using B as its similarity matrix.
4. Partition G into K disconnected subgraphs.

Theorem (KTT): Let $\hat{B}_n$ with entries $\hat{B}_n^{(ij)}$ be the affinity matrix from OSC. Then $\forall$ pairs (i,j) belonging to different communities and sparsity factor satisfying $n\rho_n = \omega\{(\log n)^{4c}\}$,

$$\max_{i,j} \hat{B}_n^{(ij)} = O_P\left(\frac{(\log n)^c}{\sqrt{n\rho_n}}\right)$$

Sparse Subspace Clustering

Corollary: The ASE of $A \sim \text{PABM}(\{\lambda^{(kl)}\}_K)$ lies near a collection of K-dimensional subspaces in K^2 dimensions.

Algorithm: Sparse Subspace Clustering (Elhamifar & Vidal, 2009):

1. Solve n optimization problems $c_i = \arg\min_c \|c\|_1$ subject to $x_i = Xc$ and $c^{(i)} = 0$.
   This is typically performed via LASSO:
   $c_i = \arg\min \frac{1}{2} \|x_i - X_{-i} c\|_2^2 + \lambda \|c\|_1$
2. Compile solutions $C = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}$.
3. Construct affinity matrix $B = |C| + |C^\top|$.

Sparse Subspace Clustering

Noroozi et al. observed that the rank of P is K^2 and the columns of P belonging to each community has rank K to justify SSC for the PABM.

$$c_i = \arg\min_c \|c\|_1 \text{ subject to } A_{\cdot,i} = Ac \text{ and } c^{(i)} = 0$$

They were able to show that this obeys SDP if we replace A with P.

GRDPG-based approach: Apply SSC to the ASE of A.

Stronger result: Apply SSC to the eigenvectors of A.

$$c_i = \arg\min_c \|c\|_1 \text{ subject to } \hat{v}_i = \hat{V} c \text{ and } c^{(i)} = 0$$

$$A \approx \hat{V} \hat{\Lambda} \hat{V}^\top$$

Sparse Subspace Clustering

Theorem (KTT):

Let

- P_n describe the edge probability matrix of the PABM with n vertices, and $A_n \sim \text{BernoulliGraph}(P_n)$.
- $\hat{V}_n$ be the matrix of eigenvectors of A_n corresponding to the $K(K+1)/2$ most positive and $K(K-1)/2$ most negative eigenvalues.

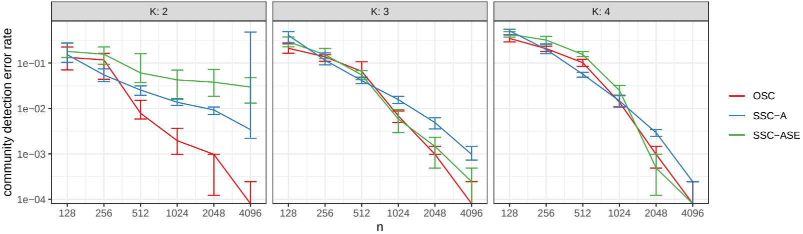
Then

- $\exists \lambda > 0$ and $N < \infty$ such that when $n > N$, $\sqrt{n} \hat{V}_n$ obeys the Subspace Detection Property with probability 1.

Remarks:

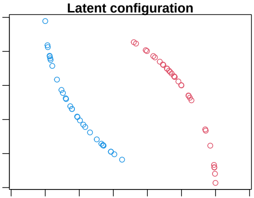
- For large n, we can identify λ for SDP (Wang and Xu, 2016).
- SDP does not guarantee community detection.

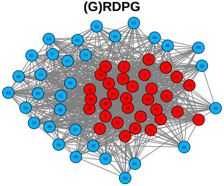
Simulation Study

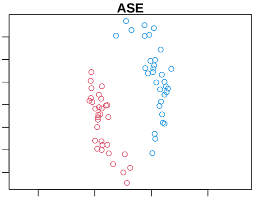


General Community Detection for the (G)RDPG

Recovery from the Adjacency Matrix







Generative Model

Let $(A, X) \sim \text{GRDPG}_{p,q}(F, n)$ as follows:

1. Define functions $\gamma_1, ..., \gamma_K$ such that each $\gamma_k : [0, 1]^r \mapsto \mathbb{R}^d$ and $\gamma_k(t) \neq \gamma_l(t)$ when $k \neq l$.
2. Sample labels $Z_1, ..., Z_n \stackrel{\text{iid}}{\sim} \text{Categorical}(\pi_1, ..., \pi_K)$.
3. Sample $T_1, ..., T_n \stackrel{\text{iid}}{\sim} D$ with support $[0, 1]^r$.
4. Set latent positions $X_i = \gamma_{Z_i}(T_i)$ and $X = \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^\top$.
5. $A \sim \text{BernoulliGraph}(X I_{p,q} X^\top)$

Examples:

1. $r = 1$ and each γ_k is a point mass $\implies$ SBM
2. $r = 1$ and the range of each γ_k is a segment from the origin to point y_k with $\|y_k\| \leq 1 \implies$ DCBM
3. $r = K$ and the range of each γ_k is the hyperplane of coordinates $x_{K(k-1)+1}$ to $x_{kK} \implies$ PABM

Community Detection

Recall that the ASE approximates the true latent positions:

$$\max_i \|\hat{X}_i - Q_n X_i\| \stackrel{a.s.}{\to} 0$$

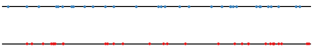
This leads to a general community detection method:

Given A, K, and d (or p and q),

1. Use ASE to approximate the latent configuration.
2. Use an appropriate clustering algorithm for the latent configuration.

Parallel Segments

Example 3: Let $U_1, ..., U_{n_1}, U_{n_1+1}, ..., U_n \overset{\text{iid}}{\sim} \text{Uniform}(0, \theta)$, $\gamma_1(t) = (t, 0)$, and $\gamma_2(t) = (t, a)$. $X_i = \gamma_1(U_i)$ for $i \leq n_1$ and $X_j = \gamma_2(U_j)$ for $n_1 + 1 \leq j \leq n$.
If we observe $X_1, ..., X_{n_1}, X_{n_1+1}, ..., X_n$, what approach will allow us to group the observations by segment?

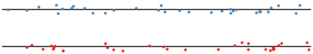


$\forall a \in (0, 1)$, $\delta \in (0, 1)$, and $K \geq 2$, $\exists N(a, \delta, K) < \infty$ such that when $\min_k n_k \geq N$, with probability at least $1 - \delta$,

1. Single linkage clustering entails perfect community detection.
2. Any ϵ-neighborhood graph with $\epsilon \leq a$ will consist of at least K disjoint subgraphs such that no subgraph contains members of two different communities.

Noisy Parallel Segments and One-Dimensional Manifolds

Example 4: Starting with the parallel segments as before, suppose instead of observing $X_1, ..., X_n$, we have noisy observations $X_1 + \xi_1, ..., X_n + \xi_n$ such that $\max_i \|\xi_i\| = \xi \leq a/3$.



Then we can derive similar statements as in Example 3.

This also holds for noisy points sampled uniformly on one-dimensional manifolds such that the manifolds are distance at least a apart.

Since the ASE of a RDPG generated from points on these segments/curves will reconstruct the original segments/curves with noise, we can apply this to the RDPG (if sufficient n).

Future Work

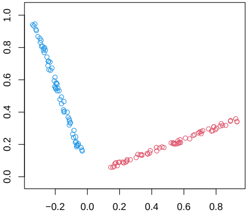
1. Derive formal statements showing that the ASE of a random graph generated by these latent configurations produces the correct conditions for community detection.
2. Extend results to non-uniform distributions.
3. Extend results to multidimensional manifolds.
4. Relax condition for the minimum distance between manifolds.
5. Extend results to the GRDPG.

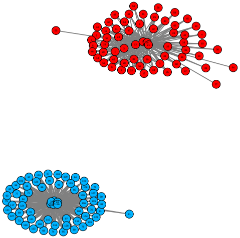
Thank you

Additional Slides

Subspace Detection Property

If X obeys the *Subspace Detection Property*, then B is sparse such that $B_{ij} = 0$ if x_i and x_j belong to different subspaces.





Theorem (Wang and Xu, 2016): If data matrix X consists of points lying close to low-dimensional subspaces such that:

1. Each point's distance to its subspace is sufficiently small.
2. The points corresponding to each subspace cover a sufficient amount of that subspace.
3. The cosine of the angles between pairs of subspaces is

Parameter Estimation

1. Let $A^{(kl)}$ be the block of edges between communities k and l.
2. For each $k, l \in [K]$, do:
   i. Compute the SVD $A^{(kl)} = U \Sigma V^\top$.
   ii. Assign $u^{(kl)}$ and $v^{(kl)}$ as the first columns of U and V. Assign $\sigma^{(kl)} \leftarrow \Sigma_{11}^{1/2}$.
   iii. Assign $\hat{\lambda}^{(kl)} \leftarrow \pm \sigma^{(kl)} u^{(kl)}$ and $\hat{\lambda}^{(kl)} \leftarrow \pm \sigma^{(kl)} v^{(kl)}$.

Theorem (KTT):

$$\max_{k,l} \|\hat{\lambda}^{(kl)} - \lambda^{(kl)}\| = O_P\Big(\frac{(\log n_k)^c}{\sqrt{n_k}}\Big)$$

Indefinite Orthogonal Group

$$\mathbb{O}(p, q) = \{Q : QI_{p,q}Q^\top = I_{p,q}\}$$

- $Q^\top Q \neq I$
- If $A \sim \mathsf{GRDPG}_{p,q}(X)$, then $A \sim \mathsf{GRDPG}_{p,q}(XQ)$
- $(Qx)^\top (Qy) = x^\top Q^\top Qy \neq x^\top y$
- $\|Q\| \neq 1 \implies \|Qx - Qy\| \neq \|x - y\|$

Non-Spectral Community Detection for Block Models

Likelihood

$$L = \prod_{i<j} \prod_{k,l}^{K} \left(p_{k,l,i,j}^{A_{ij}} (1 - p_{k,l,i,j})^{1-A_{ij}}\right)^{z_{ik} z_{jl}}$$

Example: DCBM ($p_{k,l,i,j} = \theta_{kl} \omega_i \omega_j$)

$$L = \prod_{i<j} \prod_{k,l}^{K} \left((\theta_{kl} \omega_i \omega_j)^{A_{ij}} (1 - \theta_{kl} \omega_i \omega_j)^{1-A_{ij}}\right)^{z_{ik} z_{jl}}$$

- ML method for community detection: $\hat{\vec{z}} = \arg\max_{\vec{z}} L$
- NP-complete
  - Expectation-Maximization
  - Bayesian methods
  - Spectral methods

Expectation Maximization for the PABM

Full data log-likelihood

$$
\begin{aligned}
\log L = & \sum_{i<j} \sum_{k,l} z_{ik} z_{jl} (A_{ij} \log \lambda_{il} \lambda_{jk} + (1 - A_{ij}) \log(1 - \lambda_{il} \lambda_{jk})) \\
& + \sum_{i} \sum_{k} z_{ik} \log \pi_k
\end{aligned}
$$

E-step

- $\gamma_{ik} = P(Z_i = k \mid \{\pi_l\}, \{\lambda_{jl}\})$
- $\log \gamma_{ik} \propto$
  $\log \pi_k + \sum_{j \neq i} \sum_l \pi_{jl} (A_{ij} \log \lambda_{il} \lambda_{jk} + (1 - A_{ij}) \log(1 - \lambda_{il} \lambda_{jk}))$

M-step

- $\pi_k = \frac{1}{n} \sum_i \gamma_{ik}$
- $\{\lambda_{ik}\} = \arg\max_{\{\lambda_{ik}\}} E_Z[\log L]$

MCMC Sampling for the PABM

Priors:

- $Z_i \overset{\text{iid}}{\sim} Categorical(\pi_1, ..., \pi_K)$
- $\lambda_{ik} \overset{\text{ind}}{\sim} Beta(a_{ik}, b_{ik})$

Full joint distribution:

$$\begin{aligned}\log p = constant \\ &+ \sum_{i<j} \sum_k \sum_l z_{ik} z_{jl} (A_{ij} \log \lambda_{il} \lambda_{jk} + (1 - A_{ij}) \log(1 - \lambda_{il} \lambda_{jk})) \\ &+ \sum_k \sum_i z_{ik} \log \pi_k \\ &+ \sum_i \sum_k (a_{ik} - 1) \log \lambda_{ik} + (b_{ik} - 1) \log(1 - \lambda_{ik})\end{aligned}$$

Variational Inference for the PABM

Mean Field Variational Inference

- Minimize $d_{KL}(p||q)$
  - p is the joint distribution
  - q is a density of some form
- Restrict $q(\vec{z}, \{\lambda_{ik}\}) = \Big(\prod_i q_{z_i}(z_i)\Big)\Big(\prod_{i,k} q_{\lambda_{ik}}(\lambda_{ik})\Big)$
- Iterative solution: $q_{\theta_i}^{(t+1)} \propto \exp(E_{\theta_{-i}^{(t)}}[\log p])$
- Approximate solution for the PABM
  - $Z_i \mid \{a'_{ik}\}, \{b'_{ik}\} \sim Categorical(\pi'_1, ..., \pi'_K)$
  - $\lambda_{ik} \mid \{a'_{-i,-k}\}, \{b'_{-i,-k}\}, \{\pi'_k\} \sim Beta(a'_{ik}, b'_{ik})$
  - Iteratively update $\{\pi'_K\}, \{a'_{ik}\}, \{b'_{ik}\}$ until convergence