Estimation methods in network models

Purnamrita Sarkar

Department of Statistics and Data Sciences
University of Texas, Austin

Networks in a nutshell

1. Models
    1.1 Latent variable models
        - Stochastic Blockmodels, and its variants
        - Random dot product models
        - Latent distance models

    1.2 Exponential random graph models
2. Estimation methods
    2.1 Nonconvex approaches
        - Spectral methods
        - Likelihood based methods
        - Variational inference

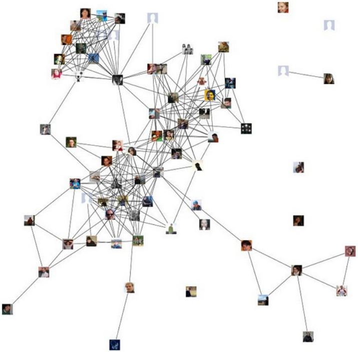
    2.2 Convex approaches - semidefinite programming

Clustering or Community detection

1. A fundamental problem in exploratory analysis
2. Communities - groups of nodes which behave similarly

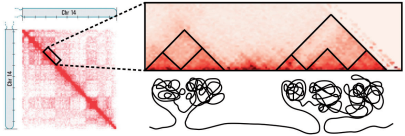
Clustering or Community detection

1. A fundamental problem in exploratory analysis
2. Communities - groups of nodes which behave similarly
   2.1 Networks: nodes are entities, links represent interactions between nodes. Communities could be
      2.1.1 groups of users in Facebook



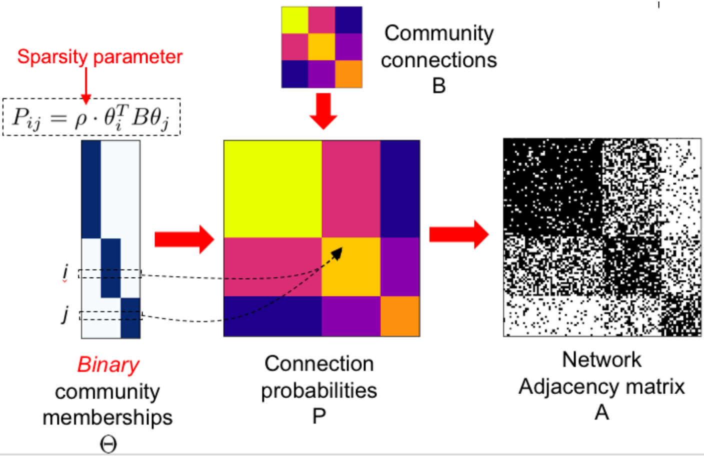
Clustering or Community detection

1. A fundamental problem in exploratory analysis
2. Communities - groups of datapoints which behave similarly
   2.1 Networks: nodes are entities, links represent interactions between nodes. Communities could be
      - positions of a chromatin which are associated via 3D looping (Weinreb and Raphael 2005, Wang, S., Ursu, Kundaje and Bickel, 2018)



      - Nodes denote a position on the chromatin
      - edges measure how close they are in a 3D arrangement
      - Goal is to find loops or hairballs, formally known as Topologically associated domains (TADs)
      - These are preserved across cell types and also different species

The stochastic block model (Holland, Laskey and Leinhardt 1983)



Sparsity parameter

$$P_{ij} = \rho \cdot \theta_i^T B \theta_j$$

Community connections B

Binary community memberships Θ

Connection probabilities P

Network Adjacency matrix A

Inference methods

1. We will start by writing down the log likelihood for a fixed Θ.
2. First note that the conditional expectation matrix $E[A|\Theta] = \Theta B \Theta^T$ is blockwise constant.

$$\begin{aligned} &\log P(A; \Theta, B) \\ &= \sum_{i,j} A_{ij} \log P_{ij} + (1 - A_{ij}) \log(1 - P_{ij}) \end{aligned}$$

Inference methods

1. We will start by writing down the log likelihood for a fixed Θ.
2. First note that the conditional expectation matrix $E[A|\Theta] = \Theta B \Theta^T$ is blockwise constant.

$$
\begin{aligned}
&\log P(A;\Theta,B)\\
&= \sum_{i,j} A_{ij}\log P_{ij} + (1-A_{ij})\log(1-P_{ij})\\
&= \sum_{i,j}\sum_{k,\ell} \underbrace{\Theta_{ik}\Theta_{j\ell}}_{1 \text{ if } i\in C_k, j\in C_\ell} \left(A_{ij}\log B_{k\ell} + (1-A_{ij})\log(1-B_{k\ell})\right)
\end{aligned}
$$

Profile Likelihood [Bickel et al, 2009]

1. We will massage the log likelihood to write:

$$\log P(A; \Theta, B)$$

Profile Likelihood [Bickel et al, 2009]

1. We will massage the log likelihood to write:

$$
\begin{aligned}
&\log P(A;\Theta,B)\\
&=\sum_{i,j}\sum_{k,\ell}\underbrace{\Theta_{ik}\Theta_{j\ell}}_{1\text{ if }i\in C_k,j\in C_\ell}\left(A_{ij}\log B_{k\ell}+(1-A_{ij})\log(1-B_{k\ell})\right)
\end{aligned}
$$

Profile Likelihood [Bickel et al, 2009]

1. We will massage the log likelihood to write:

$$
\begin{aligned}
&\log P(A; \Theta, B)\\
&= \sum_{i,j} \sum_{k,\ell} \underbrace{\Theta_{ik}\Theta_{j\ell}}_{1 \text{ if } i \in C_k, j \in C_\ell} \left(A_{ij} \log B_{k\ell} + (1 - A_{ij}) \log(1 - B_{k\ell})\right)\\
&= \sum_{k,\ell} \underbrace{O_{k\ell}}_{\substack{\#\text{ edges between}\\ \text{clusters } k \text{ and } \ell}} \log B_{k\ell} + (\underbrace{n_k}_{|C_k|} n_\ell - O_{k\ell}) \log(1 - B_{k\ell})
\end{aligned}
$$

Profile Likelihood [Bickel et al, 2009]

1. We will massage the log likelihood to write:

$$\begin{aligned}
&\log P(A;\Theta,B)\\
&=\sum_{i,j}\sum_{k,\ell}\underbrace{\Theta_{ik}\Theta_{j\ell}}_{1\text{ if }i\in C_k,j\in C_\ell}\left(A_{ij}\log B_{k\ell}+(1-A_{ij})\log(1-B_{k\ell})\right)\\
&=\sum_{k,\ell}\underbrace{O_{k\ell}}_{\substack{\#\text{ edges between}\\ \text{clusters } k \text{ and } \ell}}\log B_{k\ell}+(\underbrace{n_k}_{|C_k|}n_\ell-O_{k\ell})\log(1-B_{k\ell})\\
&=\sum_{k,\ell}O_{k\ell}\log\underbrace{\frac{O_{k\ell}}{n_kn_\ell}}_{\text{estimated parameter}}+(n_kn_\ell-O_{k\ell})\log\frac{n_kn_\ell-O_{k\ell}}{n_kn_\ell}
\end{aligned}$$

Profile Likelihood [Bickel et al, 2009]

1. We will massage the log likelihood to write:

$$
\begin{aligned}
&\log P(A; \Theta, B)\\
&= \sum_{i,j} \sum_{k,\ell} \underbrace{\Theta_{ik}\Theta_{j\ell}}_{1 \text{ if } i\in C_k, j\in C_\ell} \left(A_{ij} \log B_{k\ell} + (1 - A_{ij}) \log(1 - B_{k\ell})\right)\\
&= \sum_{k,\ell} \underbrace{O_{k\ell}}_{\substack{\#\text{ edges between}\\ \text{clusters } k \text{ and } \ell}} \log B_{k\ell} + (\underbrace{n_k}_{|C_k|} n_\ell - O_{k\ell}) \log(1 - B_{k\ell})\\
&= \sum_{k,\ell} O_{k\ell} \log \underbrace{\frac{O_{k\ell}}{n_k n_\ell}}_{\text{estimated parameter}} + (n_k n_\ell - O_{k\ell}) \log \frac{n_k n_\ell - O_{k\ell}}{n_k n_\ell}
\end{aligned}
$$

2. Maximizing this to infer Θ

Profile Likelihood optimization

1. Optimization is an NP hard problem
2. But when there is strong signal, label switching algorithms work well (Stephens et al, 2000).
3. While there are consistency guarantees about the global optima of these objective functions, the estimation methods can easily get stuck in local optima.
4. Very closely related to the Newman Girvan modularity (Newman et al, 2004).

More estimation methods

1. Let us look at a simpler model,
   1.1 all equal size clusters
   1.2 within block connection probability p and across block q
2. The log likelihood can be now written as

$$(+\text{ve const}) \times \langle \Theta\Theta^T, A \rangle + \text{another constant}$$

Optimization perspective

1. Optimization goal -

$$\arg\max_{\Theta \in \mathcal{F}} \langle \Theta\Theta^T, A \rangle,$$

with $\mathcal{F}$ being some feasible set.

Optimization perspective

1. Optimization goal -

$$\arg\max_{\Theta \in \mathcal{F}} \langle \Theta \Theta^T, A \rangle,$$

   with $\mathcal{F}$ being some feasible set.

2. Question is, what is $\mathcal{F}$?

Optimization perspective

1. Optimization goal -

$$\arg\max_{\Theta \in \mathcal{F}} \langle \Theta\Theta^T, A \rangle,$$

   with $\mathcal{F}$ being some feasible set.

2. Question is, what is $\mathcal{F}$?

3. For k equal communities, $\Theta^T\Theta = \frac{n}{k}\boldsymbol{I}$, where I is the identity matrix.

Spectral Clustering

1. Consider

$$\arg \max_{\Theta^T\Theta=\frac{n}{k}\boldsymbol{I}} \langle \Theta\Theta^T, A\rangle$$

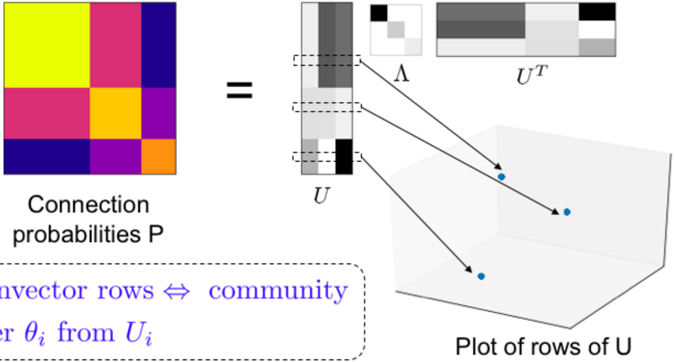
Spectral Clustering

1. Consider

$$\arg \max_{\Theta^T\Theta = \frac{n}{k} \boldsymbol{I}} \langle \Theta\Theta^T, A \rangle$$

2. As it turns out, this returns the top k eigenvectors of A (suitably scaled).
   2.1 Widely used in ML (Ng et al 2002, Shi and Malik 2001)
   2.2 Often you compute top k eigenvectors of the normalized adjacency matrix. (for consistency results, see Rohe et al 2010)

Spectral Clustering - why this works

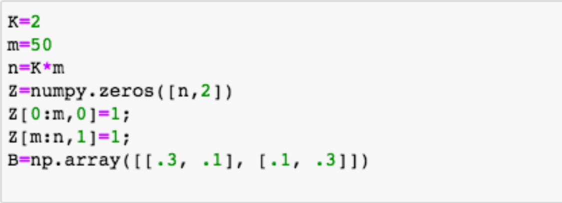


Eigenvector rows ⇔ community

$\Rightarrow$ infer θ_i from U_i

Spectral Clustering - try with code

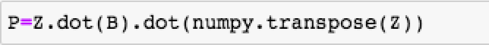
1. Lets generate a network from a blockmodel
2. First figure out parameters



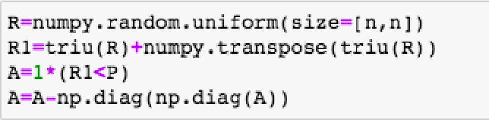
```
K=2
m=50
n=K*m
Z=numpy.zeros([n,2])
Z[0:m,0]=1;
Z[m:n,1]=1;
B=np.array([[.3, .1], [.1, .3]])
```

Spectral Clustering - try with code

1. Now build a symmetric random uniform matrix



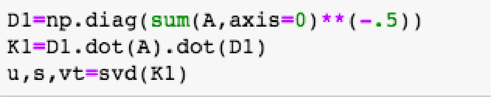
```
P=Z.dot(B).dot(numpy.transpose(Z))
```



```
R=numpy.random.uniform(size=[n,n])
R1=triu(R)+numpy.transpose(triu(R))
A=1*(R1<P)
A=A-np.diag(np.diag(A))
```

Spectral Clustering - try with code

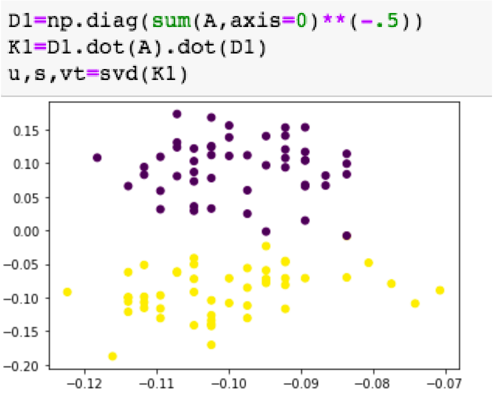
1. Do Spectral Clustering



```
D1=np.diag(sum(A,axis=0)**(-.5))
K1=D1.dot(A).dot(D1)
u,s,vt=svd(K1)
```

Spectral Clustering - try with code

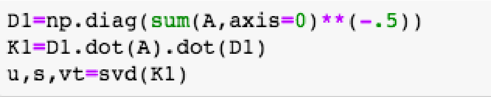
1. Do Spectral Clustering



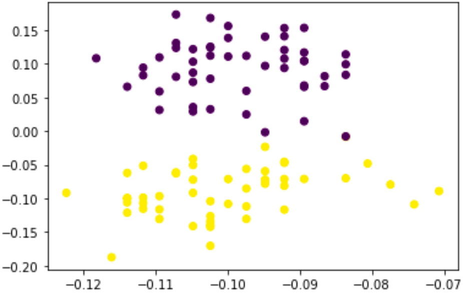
```
D1=np.diag(sum(A,axis=0)**(-.5))
K1=D1.dot(A).dot(D1)
u,s,vt=svd(K1)
```

Spectral Clustering - try with code

1. Do Spectral Clustering



```
D1=np.diag(sum(A,axis=0)**(-.5))
K1=D1.dot(A).dot(D1)
u,s,vt=svd(K1)
```



2. Accuracy of kmeans 90%

Spectral Clustering - sparse graph

1. Now generate a sparse graph with average degree about one fifth
2. How will you normalizing using degrees if there are zero degree nodes

Spectral Clustering - sparse graph

1. Now generate a sparse graph with average degree about one fifth

2. How will you normalizing using degrees if there are zero degree nodes

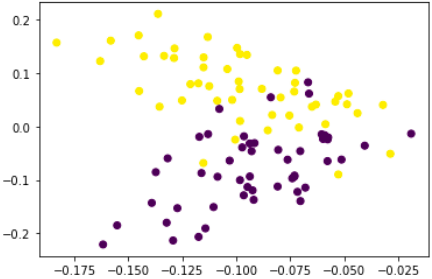
3. The trick is to add a diagonal to the degree matrix with τI, where τ is avg degree.

Spectral Clustering - sparse graph

1. Now generate a sparse graph with average degree about one fifth

2. How will you normalizing using degrees if there are zero degree nodes

3. The trick is to add a diagonal to the degree matrix with τI, where τ is avg degree.



Spectral Clustering - sparse graph

1. Why would this work?

Spectral Clustering - sparse graph

1. Run k-means on the top 2 eigenvectors of $D^{-1/2}AD^{-1/2}$

Spectral Clustering - sparse graph

1. Run k-means on the top 2 eigenvectors of $D^{-1/2}AD^{-1/2}$
2. Accuracy is about 50%

Spectral Clustering - sparse graph

1. Run k-means on the top 2 eigenvectors of $D^{-1/2}AD^{-1/2}$
2. Accuracy is about 50%
3. What if we also do the row normalization

Spectral Clustering - sparse graph

1. Run k-means on the top 2 eigenvectors of $D^{-1/2}AD^{-1/2}$
2. Accuracy is about 50%
3. What if we also do the row normalization
4. Accuracy is 90%

Spectral Clustering - sparse graph

1. Run k-means on the top 2 eigenvectors of $D^{-1/2}AD^{-1/2}$
2. Accuracy is about 50%
3. What if we also do the row normalization
4. Accuracy is 90%
5. Next – Convex relaxations

Convex relaxations

1. Recall our simple setting

2. Maximizing the log likelihood boiled down to maximizing $\arg\max_{\Theta \in \mathcal{F}} \langle \Theta\Theta^T, A \rangle$

Convex relaxations

1. Recall our simple setting
2. Maximizing the log likelihood boiled down to maximizing $\arg\max_{\Theta \in \mathcal{F}} \langle \Theta\Theta^T, A \rangle$
3. Natural feasible set is

$$\begin{gathered} \mathcal{F} = \{\Theta_{ia} \in [0,1], \forall i \in [n], a \in [k] \\ \sum_a \Theta_{ia} = 1, \forall i \in [n]\} \end{gathered}$$

4. Instead of the above nonconvex objective, we will consider:

$$\arg\max_{X \in \mathcal{F}'} \langle X, A \rangle$$

Convex relaxations

1. Recall our simple setting
2. Maximizing the log likelihood boiled down to maximizing $\arg\max_{\Theta \in \mathcal{F}} \langle \Theta\Theta^T, A \rangle$
3. Natural feasible set is

$$\mathcal{F} = \{\Theta_{ia} \in [0,1], \forall i \in [n], a \in [k]$$
$$\sum_a \Theta_{ia} = 1, \forall i \in [n]\}$$

4. Instead of the above nonconvex objective, we will consider:

$$\arg\max_{X \in \mathcal{F}'} \langle X, A \rangle$$

This is a convex objective function as long as we are careful about $\mathcal{F}'$

Convex relaxations

1. We can think of the ideal X as a clustering matrix

$$X_{ij} = \begin{cases} 1 & \text{if } i, j \text{ belong to same class} \\ 0 & \text{otherwise} \end{cases}$$

2. We can use a slightly different feasible set, namely

$$\begin{aligned} \mathcal{F}' = \{ & X_{ij} \in [0,1], \forall i, j \in [n] \\ & X_{ii} = 1 \\ & \sum_j X_{ij} = 1, \forall i \in [n] \\ & X \succeq 0 \} \end{aligned}$$

Semidefinite relaxations -pros

1. Variety of feasible sets for blockmodels and degree corrected blockmodels
    1.1 Guéndon and Vershynin 2015, Amini and Levina 2017, Yan and S. 2018, Perry and Wein 2015, Chen et al 2012, 2015

Semidefinite relaxations -pros

1. Variety of feasible sets for blockmodels and degree corrected blockmodels
   1.1 Guéndon and Vershynin 2015, Amini and Levina 2017, Yan and S. 2018, Perry and Wein 2015, Chen et al 2012, 2015
2. Robust to outliers (Cai et al 2014, Yan and S. 2016)

Semidefinite relaxations -pros

1. Variety of feasible sets for blockmodels and degree corrected blockmodels
    1.1 Guéndon and Vershynin 2015, Amini and Levina 2017, Yan and S. 2018, Perry and Wein 2015, Chen et al 2012, 2015
2. Robust to outliers (Cai et al 2014, Yan and S. 2016)
3. Has superior performance for sparse networks (Guedon et al 2014)

Semidefinite relaxations

1. Very slow, scales to a few thousands of nodes

Semidefinite relaxations

1. Very slow, scales to a few thousands of nodes
2. Requires to store $n \times n$ clustering matrix–may become prohibitive for large networks
   2.1 Recently there have been a lot of effort on optimizing quantities like $\langle A, YY^T \rangle$
   2.2 Known as Burer Monteiro methods, these have been shown to enjoy nice theoretical properties, e.g. the local optima in fact are the global optima, and saddle points can be escaped [Mei et al, 2017, Boumal et al 2018].

Semidefinite relaxations

1. Very slow, scales to a few thousands of nodes
2. Requires to store $n \times n$ clustering matrix–may become prohibitive for large networks
   2.1 Recently there have been a lot of effort on optimizing quantities like $\langle A, YY^T \rangle$
   2.2 Known as Burer Monteiro methods, these have been shown to enjoy nice theoretical properties, e.g. the local optima in fact are the global optima, and saddle points can be escaped [Mei et al, 2017, Boumal et al 2018].
3. Next–Bayesian methods

Bayesian methods

1. MCMC type methods scale poorly to very large networks.
2. Variational approximations provide fast iterative algorithms
    - 2.1 The variational principle gives us a "tractable" lower bound of the observed data likelihood, i.e. $P(A; B) = \sum_{\Theta} P(A; \Theta, B)$

Bayesian methods

1. MCMC type methods scale poorly to very large networks.

2. Variational approximations provide fast iterative algorithms

    2.1 The variational principle gives us a “tractable” lower bound of the observed data likelihood, i.e. $P(A; B) = \sum_{\Theta} P(A; \Theta, B)$

    2.2 Good news:

    - flexible to any alteration to the model like added covariates
    - global optima is theoretically consistent (Celisse et al. 2012, Bickel et al. 2013)

    2.3 (Some) bad news:

    - Bad local optima (S. et al, 2018), but there is hope to escape these by making small changes to the algorithm
    - Initializing close to the global optima helps. (Zhang and Zhou, 2017)

Variational likelihood

1. The complete likelihood for the SBM is given by

   $$P(A, \Theta; B, \pi) = \prod_{i<j} \prod_{a,b} (B_{ab}^{A_{ij}} (1 - B_{ab})^{1-A_{ij}})^{\Theta_{ia}\Theta_{jb}} \prod_{i} \prod_{a} \pi_a^{\Theta_{ia}}.$$

   Integrating out Θ, the data likelihood is

   $$P(A; B, \pi) = \sum_{\Theta \in \times} P(A, \Theta; B, \pi),$$

   where $\times$ is the space of all $n \times K$ matrices with exactly one 1 in each row.

2. Need a tractable form of the likelihood

Variational likelihood

$$\begin{aligned}\log P(A; B, \pi) &= \log\left(\sum_{\Theta} \frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \psi(\Theta)\right) \\ &\overset{\text{(Jensen)}}{\geq} \sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta) \qquad \forall \psi \text{ prob. on } \Theta.\end{aligned}$$

Variational likelihood

$$\begin{aligned}\log P(A; B, \pi) &= \log \left(\sum_{\Theta} \frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \psi(\Theta) \right) \\ &\overset{\text{(Jensen)}}{\geq} \sum_{\Theta} \log \left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \right) \psi(\Theta) \qquad \forall \psi \text{ prob. on } \Theta.\end{aligned}$$

1. This is very similar to for positive numbers $x_1, \ldots, x_n$

$$\log \left(\frac{\sum_i x_i}{n} \right) \geq \sum_i \frac{\log x_i}{n}.$$

2. Why is that?

AM/GM inequality

1. This is very similar to for positive numbers $x_1, \ldots, x_n$

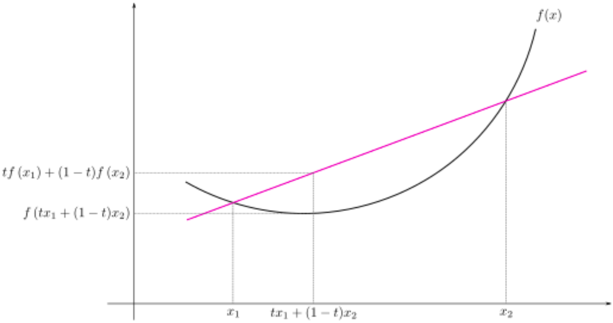
$$\log\left(\frac{\sum_i x_i}{n}\right) \geq \sum_i \frac{\log x_i}{n}.$$

2. Why is that?

Jensen's inequality

1. More generally, we see that $E[f(X)] \geq f(E[X])$ for convex function f

2. Here log is a concave function



Variational likelihood

$$
\begin{aligned}
\log P(A; B, \pi) &= \log \left(\sum_{\Theta} \frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \psi(\Theta) \right) \\
&\overset{\text{(Jensen)}}{\geq} \sum_{\Theta} \log \left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \right) \psi(\Theta) \qquad \forall \psi \text{ prob. on } \times .
\end{aligned}
$$

Variational likelihood

$$\begin{aligned}\log P(A; B, \pi) &= \log\left(\sum_{\Theta} \frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \psi(\Theta)\right) \\ &\overset{\text{(Jensen)}}{\geq} \sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta) \qquad \forall \psi \text{ prob. on } \times.\end{aligned}$$

1. Equality holds for $\psi^*(\Theta) = P(\Theta | A; B, \pi)$.

$$\log P(A; B, \pi) = \max_{\psi \in \Psi} \sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta).$$

Variational likelihood

$$
\begin{aligned}
\log P(A; B, \pi) &= \log\left(\sum_{\Theta} \frac{P(A, \Theta; B, \pi)}{\psi(\Theta)} \psi(\Theta)\right) \\
&\overset{\text{(Jensen)}}{\geq} \sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta) \qquad \forall \psi \text{ prob. on } \times.
\end{aligned}
$$

1. Equality holds for $\psi^*(\Theta) = P(\Theta | A; B, \pi)$.

$$
\log P(A; B, \pi) = \max_{\psi \in \Psi} \sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta).
$$

2. Replace Ψ above by some subclass Ψ_0,

$$
\log P(A; B, \pi) \geq \max_{\psi \in \Psi_0 \subset \Psi} \underbrace{\sum_{\Theta} \log\left(\frac{P(A, \Theta; B, \pi)}{\psi(\Theta)}\right) \psi(\Theta)}_{\text{ELBO, variational llh}}.
$$

Variational likelihood - mean field approximation

1. Here is the mean-field lower bound

$$\Psi_{MF} \equiv \{\psi : \psi(z_1, \dots, z_n) = \prod_{j=1}^{n} \psi_j(z_j)\}.$$

$$\Rightarrow \ell_{MF}(\psi, B, \pi) = \sum_{i<j,a,b} \psi_{ia}\psi_{jb}(A_{ij} \log B_{ab} + (1 - A_{ij}) \log(1 - B_{ab})) - \mathtt{KL}(\psi || \pi^{\otimes n})$$

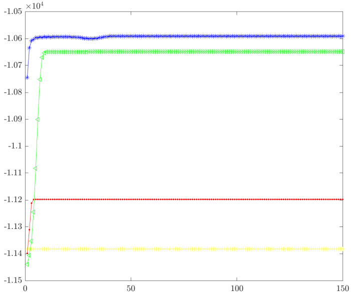
Variational likelihood - Alternating algorithm

1. Coordinate ascent, alternate between MF parameters and model parameters
   1.1 $\max_{\psi} \ell_{MF}(\psi, B, \pi)$

$$\begin{aligned} \text{subject to} \sum_{a} \psi_{ia} &= 1, \ \text{for all } 1 \leq i \leq n \\ \psi_{ia} &\geq 0, \ \text{for all } 1 \leq i \leq n, 1 \leq a \leq K, \end{aligned}$$

   1.2 $\max_{B,\pi} \ell_{MF}(\psi, B, \pi)$

Problem of local optimum



1. $K = 3,\ B = 0.5 \cdot \begin{bmatrix} 1 & 0.4 & 0.1 \\ 0.4 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix},\ \pi = (1/3, 1/3, 1/3),\ n = 600.$

2. truth; $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; $\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; (1/3,1/3,1/3)

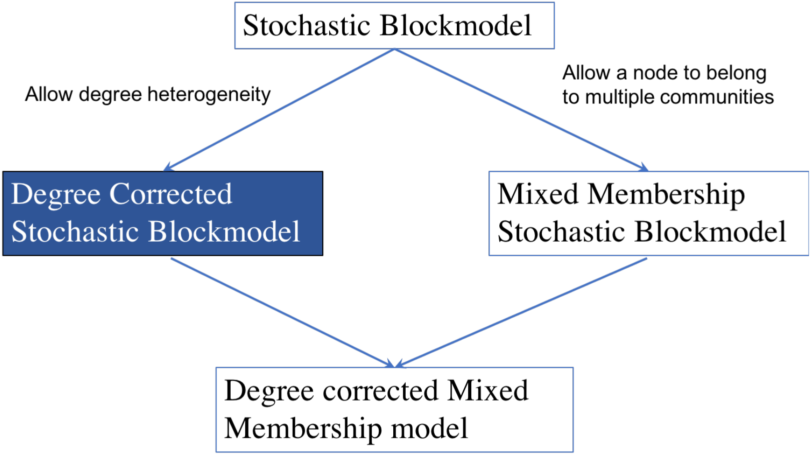
Problem of local optimum

1. Random initializations do not work well. You can get stuck in a local optima.
2. Need to initialize carefully, or regularize.
   - 2.1 Initialize the model parameters with reasonable estimates
   - 2.2 Skip the parameter update step
   - 2.3 This forces the algorithm to not veer off towards bad local optima, and as a result, with a constant probability, a random initialization can get to the global optima.

Problem of local optimum

1. Random initializations do not work well. You can get stuck in a local optima.
2. Need to initialize carefully, or regularize.
   2.1 Initialize the model parameters with reasonable estimates
   2.2 Skip the parameter update step
   2.3 This forces the algorithm to not veer off towards bad local optima, and as a result, with a constant probability, a random initialization can get to the global optima.
3. Next – Beyond blockmodels

Generalizations of a blockmodel



Stochastic Blockmodel

Allow degree heterogeneity

Allow a node to belong
to multiple communities

Degree Corrected
Stochastic Blockmodel

Mixed Membership
Stochastic Blockmodel

Degree corrected Mixed
Membership model

A real clustering dataset

1. Lets talk about the political blogs dataset.
2. Here, every node is a political blog, a link signifies which blog points to which other blog.
3. The labels (blue and red) signify political orientation of the blogs

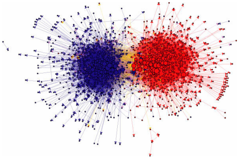


Figure: Lada Adamic and Natalie Glance."The political blogosphere and the 2004 US election: divided they blog." Proceedings of the 3rd international workshop on Link discovery. ACM, 2005.

Political blogs degree distribution

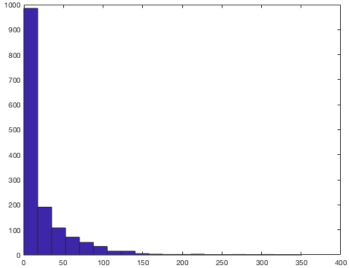


Figure: Histogram of degrees of nodes (after removing directions on edges)

Political blogs degree distribution

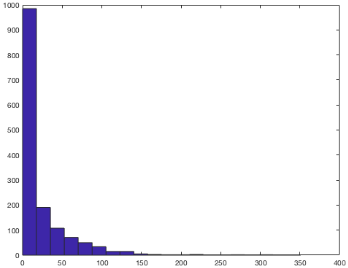


Figure: Histogram of degrees of nodes (after removing directions on edges)

1. Spectral Clustering using the top 2 eigenvectors of A fails here – clustering accuracy 60%

Degree-corrected SBM (Karrer, Newman 2010)

Degree homogeneity in SBM models

1. Expected degrees are equal among different nodes

2. Real networks, there are usually “hub” - nodes with very large degrees

Degree-corrected SBM (Karrer, Newman 2010)

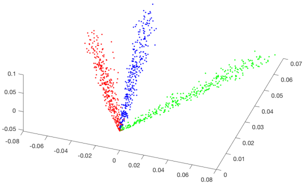
Degree homogeneity in SBM models

1. Expected degrees are equal among different nodes
2. Real networks, there are usually "hub" - nodes with very large degrees

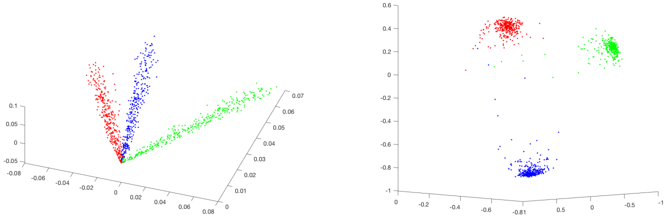
Easy to fix-

1. Add degree parameter to each node that encodes the popularity
2. $P(A_{ij} = 1) = \rho_n \gamma_i \gamma_j \theta_i^T B \theta_j$, where γ_i is the degree parameter of node i.
3. Put constraints on the sum of them to make things identifiable.

Methods and related work



Methods and related work



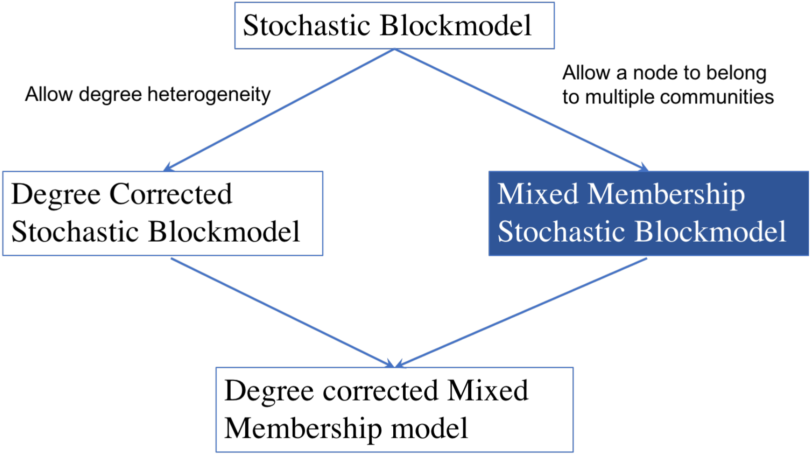
Un-normalized and Row normalized top K eigenvectors

1. Normalize top K eigenvectors and do clustering (Chaudhuri et al 2012, Qin et al 2013)
2. k-median based clustering algorithm on a low rank approximation of A followed by a refinement procedure. (Gao et al 2016).
3. SDP-based methods with regularization (Chen et al 2017)

Political blogs again

1. If we take the top eigenvectors of the adjacency matrix and row normalize them prior to kmeans, the accuracy is around 80%
2. If we just do normalized Spectral clustering on the largest connected component, then the error is nearly 50% (whether we row normalize or not)
3. If we do regularized spectral clustering without row normalization, error is about 30%.
4. If we do regularized spectral clustering with row normalization, error is about 5%.

Generalizations of SBM



Stochastic Blockmodel
Allow degree heterogeneity
Allow a node to belong to multiple communities
Degree Corrected Stochastic Blockmodel
Mixed Membership Stochastic Blockmodel
Degree corrected Mixed Membership model

Methods and related work

1. SDP's cannot be extended easily to this settings.

Methods and related work

1. SDP's cannot be extended easily to this settings.
2. Variational inference (Airoldi et al 2008, Gopalan et al 2013)

Methods and related work

1. SDP's cannot be extended easily to this settings.
2. Variational inference (Airoldi et al 2008, Gopalan et al 2013)
3. Tensor based methods (Anandkumar 2014, Hopkins et al 2018)

Methods and related work

1. SDP's cannot be extended easily to this settings.
2. Variational inference (Airoldi et al 2008, Gopalan et al 2013)
3. Tensor based methods (Anandkumar 2014, Hopkins et al 2018)
4. If B is positive semidefinite, then one can pose this as a symmetric non-negative matrix factorization problem (SNMF).
   - 4.1 Bayesian variant of NMF (Psorakis 2011)
   - 4.2 Use geometric intuition to solve the SNMF problem (Mao, S. and Chakrabarti 2017).

Methods and related work

1. SDP's cannot be extended easily to this settings.
2. Variational inference (Airoldi et al 2008, Gopalan et al 2013)
3. Tensor based methods (Anandkumar 2014, Hopkins et al 2018)
4. If B is positive semidefinite, then one can pose this as a symmetric non-negative matrix factorization problem (SNMF).
   - 4.1 Bayesian variant of NMF (Psorakis 2011)
   - 4.2 Use geometric intuition to solve the SNMF problem (Mao, S. and Chakrabarti 2017).
5. For degree corrected mixed membership models, one needs to adapt the Spectral algorithms further (see Zhang and Levina 2014, Jin et al 2017, Mao et al 2019).

The Mixed Membership Blockmodel (Airoldi et al, 2008)

1. Number of communities K
2. $K \times K$ matrix of connection probabilities $\mathbf{B}$
3. $\theta_i \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_K)$
4. $A_{ij} \sim E[A_{ij}|\theta] = \rho_n \theta_i^T B \theta_j =$ Call this P
5. Special case : Stochastic blockmodel when $\alpha_a \to 0$
    5.1 All $\theta_i \in \{0, 1\}^K$ have exactly one 1.
6. Large $\alpha_a \equiv$ more overlap and small $\alpha_a \equiv$ less overlap
7. Goal: Given **A**, infer $\{\theta_i\}$ and **B**

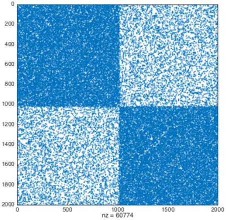
Dirichlet distribution

1. Parameters: $\alpha_1, \dots, \alpha_K > 0$
2. Density $f(x_1, \dots, x_K) = \dfrac{\prod_i x_i^{\alpha_i - 1}}{B(\alpha)}$
3. Where $x_1, \dots x_k \geq 0$ belong to the $K - 1$ simplex, i.e.

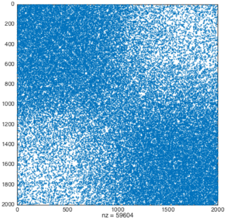
$$\sum_i x_i = 1, x_i \geq 0$$

`https://upload.wikimedia.org/wikipedia/commons/thumb/5/54/LogDirichletDensity-alpha_0.3_to_alpha_2.0.gif/500px-LogDirichletDensity-alpha_0.3_to_alpha_2.0.gif`

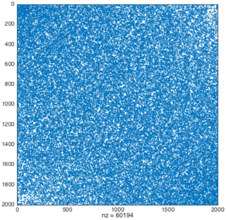
The Mixed Membership Stochastic Blockmodel (MMSB)



$\boldsymbol{\alpha} = (.005, .005)$



$\boldsymbol{\alpha} = (.2, .2)$

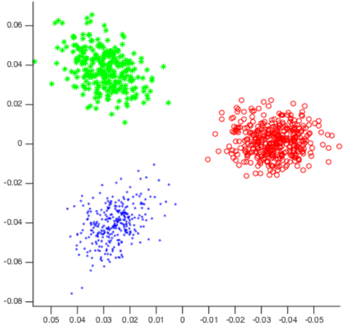


$\boldsymbol{\alpha} = (2, 2)$

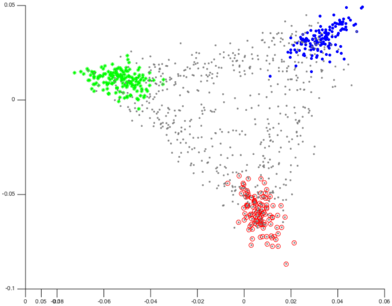
$\boldsymbol{\alpha}$ is the Dirichlet parameter for θ_i

$\boxed{\text{Given } \mathbf{A}\text{, infer } \{\theta_i\} \text{ and } \mathbf{B}}$

Eigenvectors for 3 blocks



Stochastic Blockmodel



MMSB with $\alpha = (.2, .2, .2)$

1. Highlighted are $S = \{i : \max_a \theta_{ia} > .9\}$.
2. These are "pure" nodes

Methods and related work

1. Notable methods include Variational inference (Airoldi et al 2008, Gopalan et al 2013)

2. Tensor based methods (Anandkumar 2014, Hopkins et al 2018)

3. If B is positive semidefinite, then one can pose this as a symmetric non-negative matrix factorization problem (SNMF).

   3.1 Bayesian variant of NMF (Psorakis 2011)

   3.2 Use geometric intuition to solve the SNMF problem (Mao, S. and Chakrabarti 2017).

Building the geometric intuition

1. Eigenvectors fall on a simplex
   - We are essentially looking for a way to learn with K simplexes in K dimensional space

Building the geometric intuition

1. Eigenvectors fall on a simplex

   - We are essentially looking for a way to learn with K simplexes in K dimensional space
   - All points are convex combinations of the corners

Building the geometric intuition

1. Eigenvectors fall on a simplex

    - We are essentially looking for a way to learn with K simplexes in K dimensional space

    - All points are convex combinations of the corners

    - Once you find the corners, all the parameters can be learned using a simple regression step

2. Let us try some simple ideas to find corners.

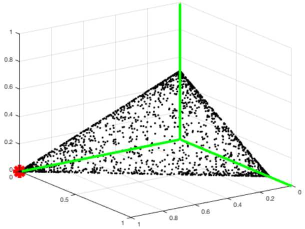
3. What if I find the node with maximum length?

    3.1 Indeed, it gives you“a nearly pure node” (with high probability).

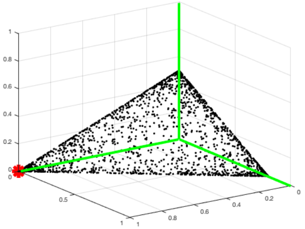
Building the geometric intuition

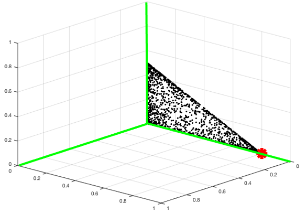
1. Scalable methods (Gillis et al 2014) in computational geometry to find corners of a noisy simplex with K corners in K dimensions.
   1.1 Find a node with largest ℓ_2 norm
   1.2 Remove its projection from the other rows.
   1.3 Repeat for K times.

Building the geometric intuition with eigenvectors of P

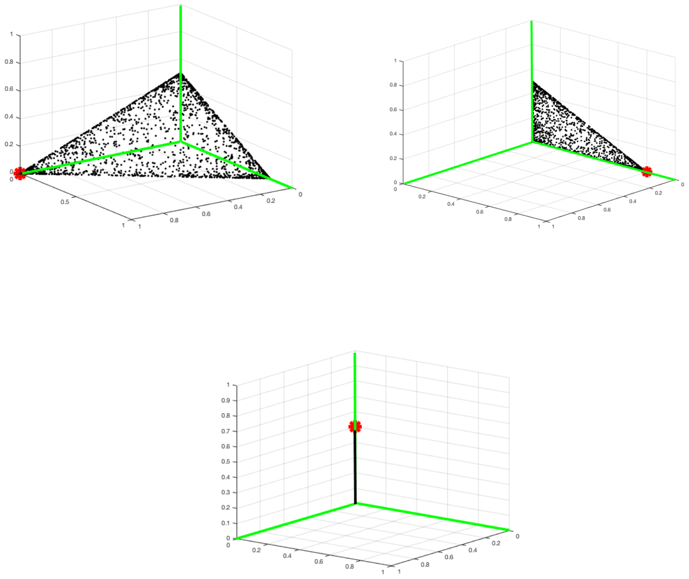


Building the geometric intuition with eigenvectors of P





Building the geometric intuition with eigenvectors of P



Putting everything together

1. Let V be eigenvectors of P
2. Let S denote the set of pure nodes
3. As it turns out, in the mixed membership model, we have:

$$V = \Theta V_S$$

Putting everything together

1. Let V be eigenvectors of P
2. Let S denote the set of pure nodes
3. As it turns out, in the mixed membership model, we have:

$$V = \Theta V_S$$

4. Now, if we estimate the pure nodes by a set $\hat{S}$, how do we get back to Θ?

Putting everything together

1. Let V be eigenvectors of P
2. Let S denote the set of pure nodes
3. As it turns out, in the mixed membership model, we have:

$$V = \Theta V_S$$

4. Now, if we estimate the pure nodes by a set $\hat{S}$, how do we get back to Θ?

Putting everything together

1. Simple: use

$$\hat{V} = \hat{\Theta} \hat{V}_{\hat{S}} \Rightarrow \hat{\Theta} = \hat{V} \hat{V}_{\hat{S}}^{-1} \tag{1}$$

2. Recall

$$\Theta B \Theta^T = VEV^T \Rightarrow B = (\Theta^T \Theta)^{-1} \Theta^T VEV^T \Theta (\Theta^T \Theta)^{-1}$$

Putting everything together

1. Simple: use

$$\hat{V} = \hat{\Theta}\hat{V}_{\hat{S}} \Rightarrow \hat{\Theta} = \hat{V}\hat{V}_{\hat{S}}^{-1} \tag{1}$$

2. Recall

$$\Theta B \Theta^T = VEV^T \Rightarrow B = (\Theta^T\Theta)^{-1}\Theta^T VEV^T \Theta(\Theta^T\Theta)^{-1}$$

3. But $V_S = (\Theta^T\Theta)^{-1}\Theta^T V$, from Eq 1.

Putting everything together

1. Simple: use

$$\hat{V} = \hat{\Theta}\hat{V}_{\hat{S}} \Rightarrow \hat{\Theta} = \hat{V}\hat{V}_{\hat{S}}^{-1} \tag{1}$$

2. Recall

$$\Theta B \Theta^T = VEV^T \Rightarrow B = (\Theta^T\Theta)^{-1}\Theta^T VEV^T \Theta(\Theta^T\Theta)^{-1}$$

3. But $V_S = (\Theta^T\Theta)^{-1}\Theta^T V$, from Eq 1.

$$\hat{B} = \hat{V}_{\hat{S}}\hat{E}\hat{V}_{\hat{S}}^T$$

Estimation in Random Dot Product Graphs (RDPG models)

1. Edge probabilities [Young and Scheinerman 2007]

$$P(A_{ij} = 1 \mid Y) = \langle \underbrace{X_i}_{\text{Latent positions}}, X_j \rangle$$

2. The generalized RDPG model [Rubin-Delanchy et al, 2017] encompasses the Stochastic Blockmodel and its variants

3. Popular methods include network embedding approaches using the adjacency matrix and its variants (Sussman et al 2012, Fishkind et al 2013, Tang et al 2013, Le et al 2017,Athreya et al 2016).

Estimation in Random Dot Product Graphs (RDPG models)

1. If we can just use Spectral Clustering, why go into all the trouble to do all we did for MMSB?
2. We could have just used $\hat{X} = \hat{V}\hat{E}^{1/2}$
3. But note that, $\hat{X}$ by itself does not mean anything. It can be used to cluster nodes for clustering models like blockmodels.
4. But when there is mixed memberships, $\hat{X}$ is essentially a transformed version of Θ.
5. But in order to get to Θ, we need to do more work.

Triangle formation and block models

1. Real social networks have many triangles, even if they are sparse.
2. To be particular, the global clustering coefficient, defined as number of triangles divided by number of closed or open triplets is often used to measure “clustered” networks are.
3. But for blockmodels or its variants, as the network gets sparser, the network becomes more treelike.

Triangle formation and block models

1. Real social networks have many triangles, even if they are sparse.
2. To be particular, the global clustering coefficient, defined as number of triangles divided by number of closed or open triplets is often used to measure "clustered" networks are.

Latent distance models

1. Latent distance models model homophily or transitivity by introducing a latent space where nodes lie.
2. Two nodes close in the latent space are more likely to be connected.
3. Why do you think this leads to transitivity and reciprocity (what is that)?

Latent distance models

1. Reciprocity:
   1.1 If $i \to j$, then the event $j \to i$ is more likely.
   1.2 Why?

Latent distance models

1. Reciprocity:
   1.1 If $i \rightarrow j$, then the event $j \rightarrow i$ is more likely.
   1.2 Why?
2. Transitivity:
   2.1 If $i \rightarrow j$ and $j \rightarrow k$, then $i \rightarrow k$ is more likely.
   2.2 Why?

Latent Distance Models [Hoff et al 2002]

1. Log likelihood:

$$\log P(A \mid \eta) = \sum_{i \neq j} \{\eta_{ij} \cdot A_{ij} - \log(1 + e^{\eta_{ij}})\},$$

$$\text{where } \eta_{ij} = \log\ odds(A_{ij}, z_i, z_j) = \alpha - \|z_i - z_j\|_2$$

2. Two stage approaches, which initialize with Spectral methods [S. and Moore, 2005] – no guarantee for global optima

3. Recently there has been some work on consistency of convex relaxation based inference, and non-convex inference methods [Ma et al 2020].

Estimation for Latent Space Models: Bayesian Approach

1. Log likelihood:

$$\log P(A \mid \underbrace{z_i, i \in [n]}_{\text{Latent positions}}) = \sum_{i \neq j} \{\eta_{ij} \cdot A_{ij} - \log(1 + e^{\eta_{ij}})\},$$

$$\text{where } \eta_{ij} = \log\ odds(A_{ij}, z_i, z_j) = \alpha - \|z_i - z_j\|_2$$

2. Alternatively, place priors on α, Z.

3. Use Metropolis-Hastings to update Z and α serially.

4. The Bayesian approach can be computationally prohibitive for larger networks.