# Scaling MMSB to Large Datasets

Akshat Jindal(150075) Shivam Utreja(150682) Pratyush Garg(150521)

IIT Kanpur

April 25, 2019

# Overview

- Relational data is ubiquitious nowadays and models to handle such data are thus important to study. In this project, we explore Mixed Membership Stochastic Blockmodels in detail starting from scratch and exploring ways to achieve scalability over baseline implementations.

# Overview

- Relational data is ubiquitious nowadays and models to handle such data are thus important to study. In this project, we explore Mixed Membership Stochastic Blockmodels in detail starting from scratch and exploring ways to achieve scalability over baseline implementations.
- We explore and implement a Nested VI algorithm to achieve faster convergence and scalability over naive inference procedures for MMSB.

# Overview

- Relational data is ubiquitious nowadays and models to handle such data are thus important to study. In this project, we explore Mixed Membership Stochastic Blockmodels in detail starting from scratch and exploring ways to achieve scalability over baseline implementations.
- We explore and implement a Nested VI algorithm to achieve faster convergence and scalability over naive inference procedures for MMSB.
- We also explore and implement a subsampling based approach to achieve scalability.

# Overview

- Relational data is ubiquitious nowadays and models to handle such data are thus important to study. In this project, we explore Mixed Membership Stochastic Blockmodels in detail starting from scratch and exploring ways to achieve scalability over baseline implementations.
- We explore and implement a Nested VI algorithm to achieve faster convergence and scalability over naive inference procedures for MMSB.
- We also explore and implement a subsampling based approach to achieve scalability.
- To better report our results, we implement the baseline approach and a few scalable approaches for a qualitative comparison.

# Previous Work

- **Stochastic Blockmodels** : The stochastic blockmodel is an extension or adaptation of mixture models. In that model, each object(data point $x_n$ , belongs to a cluster and the relationships between objects are governed by the corresponding pair of clusters.
- **Limitation** : Here, each object can only belong to one cluster, or in other words, play a single latent role.
- When a protein or a social actor interacts with different partners, different functional or social contexts may apply.

The full generative model looks like

- For $n = 1, \ldots, N$

$$z_n \sim \text{multinoulli}(\pi)$$

- For $n = 1, \ldots, N$
  - For $m = 1, \ldots, n-1$

$$A_{nm} \sim \text{Bernoulli}(\eta_{z_n, z_m})$$

# Mixed Membership Stochastic Block Model

## Setting

- Graph $\mathcal{G} = (\mathcal{N}, \mathbf{Y})$ where $\mathcal{N}$ is the number of vertices.
- $\mathbf{Y}$ : $\mathcal{N}x\mathcal{N}$ adjacency matrix
- K factions modelled by a $Kx1$ mixture vector for every vertex and context dependent latent vectors for each vertex.
- Probabilities of interactions between different factions : $\mathcal{B}$

## Generative Story

1. Draw a K dimensional mixed membership vector $\pi_{\mathbf{p}} \sim \text{Dirichlet}(\alpha) \quad \forall p \ \in \mathcal{N}$
2. For each pair of vertices $(p, q) \in \mathcal{G}$:
   - Draw membership indicator for the initiator : $z_{p->q} \sim \text{Multinomial}(\pi_p)$
   - Draw membership indicator for the initiator : $z_{q->p} \sim \text{Multinomial}(\pi_q)$
   - Sample value of interaction : $Y(p, q) \sim \text{Bernoulli}(B_{z_{p->q}, z_{q->p}})$

# Sparsity Parameter : $\rho$

- Adjacency matrices are often sparse.
- These non interactions or zeros in the matrix can either be due to rarity of interactions inherent in the data, or they may be an indication that the pair of relevant blocks rarely interacts.
- The $2^{nd}$ form of sparsity is captured by MMSB as it infers faction-faction interation strength $\mathbf{B}$. $1^{st}$ form is due to our data.
- Sparsity parameter $\rho \in [0, 1]$ is introduced.
- we down-weight the probability of successful interaction by $(1 - \rho)$ where the weight $\rho$ captures the portion of zeros that should not be explained by the blockmodel $\mathbf{B}$.

# Inference

Parameters to be estimated : $\{\alpha, z_{p->q}s, \pi_{\mathbf{p}}s, \mathbf{B}\}$. The authors use a Variational EM approach and estimate posterior distributions for $\pi_{\mathbf{p}}$s and $z_{p->q}$s using Mean-Field VB, and get point estimates for $\Theta : \mathbf{B}$ and $\alpha$.

# Posterior Inference : The E-Step

- We'll maximise $\mathcal{L}(q, \Theta^{old})$ in the E-Step wrt q, which gives $q^{opt}$ as the joint CP of $\pi_{\mathbf{p}}$s and $z_{p->q}$s.
- Since, this is intractable to calculate, we use Mean-Field VB to get $q^{opt}$
- Variational distribution taken :

$$q(\pi, \mathbf{Z}|\gamma, \phi) = \prod_{p} q_1(\pi_{\mathbf{p}}|\gamma_p) \prod_{p,q} q_2(z_{p->q}|\phi_{p->q}) q_2(z_{q->p}|\phi_{q->p})$$

  where $q_1$ is Dirichlet and $q_2$ is Multinomial. Thus the variational parameters to be inferred : $\{\gamma, \phi\}$
- Solving via the Mean-Field approach taught in class, the update equations for the parameters :

$$\phi^{new}_{p->q,g} \propto e^{\mathbf{E_q}[log\,\pi_{p,g}]} \prod_{h} ((\mathbf{B}(g,h))^{Y(p,q)}(1-\mathbf{B}(g,h))^{1-Y(p,q)})^{\phi^{old}_{q->p,h}}$$

$$\phi^{new}_{q->p,h} \propto e^{\mathbf{E_q}[log\,\pi_{q,h}]} \prod_{g} ((\mathbf{B}(g,h))^{Y(p,q)}(1-\mathbf{B}(g,h))^{1-Y(p,q)})^{\phi^{old}_{p->q,g}}$$

$$\gamma^{new}_{p,k} = \alpha_{\mathbf{k}} + \sum_{q} \phi^{new}_{p->q,k} + \sum_{q} \phi^{new}_{q->p,k}$$

for all nodes p = 1,2,...,N and k = 1,2,..,K. and g,h = 1,...,K.

# Point Estimates: The M-Step

- Using the variational distribution, we maximise the $\mathcal{L}(q^{optimal}, \Theta)$ wrt $\Theta = \{\mathbf{B}\}$
- For **B**, we have :

$$\hat{B}(g,h) = \frac{\sum_{p,q} Y(p,q) \cdot \phi_{p \to qg} \phi_{p \leftarrow qh}}{(1-\rho) \cdot \sum_{p,q} \phi_{p \to qg} \phi_{p \leftarrow qh}},$$

- The value of K is set either by using BIC as discussed in class or cross-validation if the model size is not large enough for BIC to be effective.
- $\rho$ is also estimated as $1 - \hat{d}$, where $\hat{d} = \frac{\sum_{p,q} Y(p,q)}{\mathcal{N}^2}$

# Scaling Up : Naive Inference

1. In naive inference, $\gamma$ and $\phi$ are initialized to non-informative values, and then we iterate the following two steps until convergence:
2. 
   - Update $\phi_{p->q}$ and $\phi_{q->p}$ for all edges (p, q).
   - Update $\gamma_p$ for all nodes $p \in \mathcal{N}$
   - Now, update **B** using the updated values of $\phi$
3. B is updated only once at each iteration
4. Every $\phi^t$ uses $\phi^{t-1}$ and $\gamma^{\mathbf{t}}$
5. $NK + 2N^2K$ scalars maintained across iterations
6. Doesnt capture the dependence between $\gamma$s and **B**.
7. Slow convergence :(

# Nested VI

- Nested VI takes care of this by rescheduling the parameter updates.
- Always keep the block of free parameters, $\phi_{p->q}$ and $\phi_{q->p}$, optimized given the other variational parameters : Faster convergence
- B is updated at every pair p,q of vertices now. Dependence between B and $\gamma$ thus retained : Faster convergence
- At each variational cycle we need to allocate NK + 2K scalars only : Scalable

# Nested VI

1. initialize $\vec{\gamma}^0_{pk} = \frac{2N}{K}$ for all $p,k$
2. **repeat**
3. **for** $p = 1$ to $N$
4. **for** $q = 1$ to $N$
5. get **variational** $\vec{\phi}^{t+1}_{p\to q}$ and $\vec{\phi}^{t+1}_{p\leftarrow q} = f\ (\ Y(p,q), \vec{\gamma}^t_p, \vec{\gamma}^t_q, B^t\ )$
6. partially update $\vec{\gamma}^{t+1}_p$, $\vec{\gamma}^{t+1}_q$ and $B^{t+1}$
7. **until** convergence

5.1. initialize $\phi^0_{p\to q,g} = \phi^0_{p\leftarrow q,h} = \frac{1}{K}$ for all $g,h$
5.2. **repeat**
5.3. **for** $g = 1$ to $K$
5.4. update $\phi^{s+1}_{p\to q} \propto f_1\ (\ \vec{\phi}^s_{p\leftarrow q}, \vec{\gamma}_p, B\ )$
5.5. normalize $\vec{\phi}^{s+1}_{p\to q}$ to sum to 1
5.6. **for** $h = 1$ to $K$
5.7. update $\phi^{s+1}_{p\leftarrow q} \propto f_2\ (\ \vec{\phi}^s_{p\to q}, \vec{\gamma}_q, B\ )$
5.8. normalize $\vec{\phi}^{s+1}_{p\leftarrow q}$ to sum to 1
5.9. **until** convergence

Figure: PseudoCode

# Scaling MMSB : Method - II

- Algorithm suggested by Prem K. Gopalan and David M. Blei.
- Successfully scales MMSB to real-world social/citation/biological networks.
- Uses a combination of Mean Field VI and Stochastic Optimization to speed-up MMSB.

# Model description.

The model used here is to model assortative undirected networks. The generative story is similar to the previous model, with only subtle changes.

- For each node, draw community memberships $\theta_i$ from $Dirichlet(\alpha)$
- For each pair of nodes $i$ and $j$ such that $i < j$;
  - Draw community indicator $z_{i->j}$ from $Multinoulli(\theta_i)$.
  - Draw community indicator $z_{i<-j}$ from $Multinoulli(\theta_j)$.
  - Draw connection between them using;

$$p(y_{ij} = 1 | z_{i->j}, z_{i<-j}) = \begin{cases} \beta_{z_{i->j}} & \text{if } z_{i->j} = z_{i<-j} \\ \epsilon & \text{if } z_{i->j} \neq z_{i<-j} \end{cases} \tag{1}$$

Note that in this case, the nodes form a connection with high probability only if their latent factors are from the same community. It still supports overlapping communities, because the $z$s themselves are being drawn from a Multinoulli.

# Posterior Approximation

The slight difference in the model is reflected in the slightly different, mean-field posterior approximation;

$$q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{i<j} q(z_{i->j}|\phi_{i->j}) q(z_{i<-j}|\phi_{i<-j}) \tag{2}$$

Each variable has its own distribution which makes the model very flexible, and allows us to learn each node's community memberships.

# Posterior Updates

Given the above form of $q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, we get the following form for the ELBO;

$$
\begin{aligned}
\mathcal{L} = & \sum_k \mathrm{E}_q\left[\log p(\beta_k|\eta)\right] - \sum_k \mathrm{E}_q\left[\log q(\beta_k|\lambda_k)\right] \\
& + \sum_n \mathrm{E}_q\left[\log p(\theta_n|\alpha)\right] - \sum_n \mathrm{E}_q\left[\log q(\theta_n|\gamma_n)\right] \\
& + \sum_{a,b} \mathrm{E}_q\left[\log p(z_{a\to b}|\theta_a)\right] + \mathrm{E}_q\left[\log p(z_{a\leftarrow b}|\theta_b)\right] \\
& - \sum_{a,b} \mathrm{E}_q\left[\log q(z_{a\to b}|\phi_{a\to b})\right] - \mathrm{E}_q\left[\log q(z_{a\leftarrow b}|\phi_{a\leftarrow b})\right] \\
& + \sum_{a,b} \mathrm{E}_q\left[\log p(y_{ab}|z_{a\to b}, z_{a\leftarrow b}, \boldsymbol{\beta})\right].
\end{aligned}
$$

**[S2]**

Figure: The ELBO

# Posterior Updates

- In the above equation, the summations corresponding to the communities and the nodes are called the global terms, and those related to the edges are called the local terms.
- Their corresponding parameters are called the global and local parameters respectively.
- Hence, $\boldsymbol{\lambda}$ and $\gamma$ become our global parameters, and $\boldsymbol{\Phi}$ become our local parameters.
- We can now perform alternating updates between the global(Global Step) and local(Local Step) parameters, until we converge to a local optimal of the ELBO.
- The local parameters are set using the current estimates of the global parameters.(They, themselves need to be updated in an alternating fashion. Precise details covered in the report.)
- The global parameters are updated using the gradient computed from the current estimate of the local parameters.

# Sampling techniques

- Note that in the above formulation, performing the global step would involve finding the gradient using $N^2$-many local parameters. $N$ can be of the order of millions.
- Hence, at each iteration, it is crucial that we sub-sample the graph's edges (i.e. local parameters) to make the algorithm tractable for real networks.
- Note that, any sampling technique will work, as long as the noisy gradient is unbiased, and its variance can be controlled.
- Also note, that this also acts as a natural way to be able to interleave data collection and model estimation.

# Sampling techniques

The author suggests several sampling techniques in the paper.

- Random pair sampling.
- Random node sampling.
- Link Sampling

# Link Sampling

- The other sampling schemes suggested above take into account, all pairs of nodes, irrespective of whether they are linked.
- This becomes increasingly redundant as the size of the graph increases. The number of links is far less than the number of non-links.
- The link sampling scheme overcomes this redundancy. However, we need to change the form of our approximation first.

$$q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{(i,j)\in links} q(z_{i->j}, z_{i<-j}|\phi_{ij}) \prod_{(i,j)\in nonlinks} q(z_{i->j}|\phi_{i->j}) q(z_{i<-j}|\phi_{i<-j}) \tag{3}$$

# Initialization and Complexity

- The authors provide an algorithm that initialises the global parameters, as well as sets the number of communities for the SVI algorithm.
- The local step for the SVI algorithm can be computed in $O(SK)$ operations.
- Note that it is not quadratic in $K$ because we are working with the assortativity assumption.
- The time for the global step is $O(NK)$ per iteration. We can further reduce this to $O(nK)$, where $n$ is the minibatch size; if we maintain distinct learning rates for each node.
- In the case of the link sampling algorithm, sampling all links together, gives a complexity of $O(MK + NK)$ per iteration. Its convergence is much faster, even without subsampling.

# Results and Implementation

- We implemented the naive and nested VI algorithms for a qualitative comparison.
- We tested them on 2 graphs : A 5*5 graph (small) and a 55*55 graph(big)
- Reconstruction : $\mathbf{E}[\mathbf{Y}(\mathbf{p},\mathbf{q})] = \phi^{\mathbf{T}}_{\mathbf{p}->\mathbf{q}}\mathbf{B}\phi_{\mathbf{q}->\mathbf{p}}$

# Small Graph

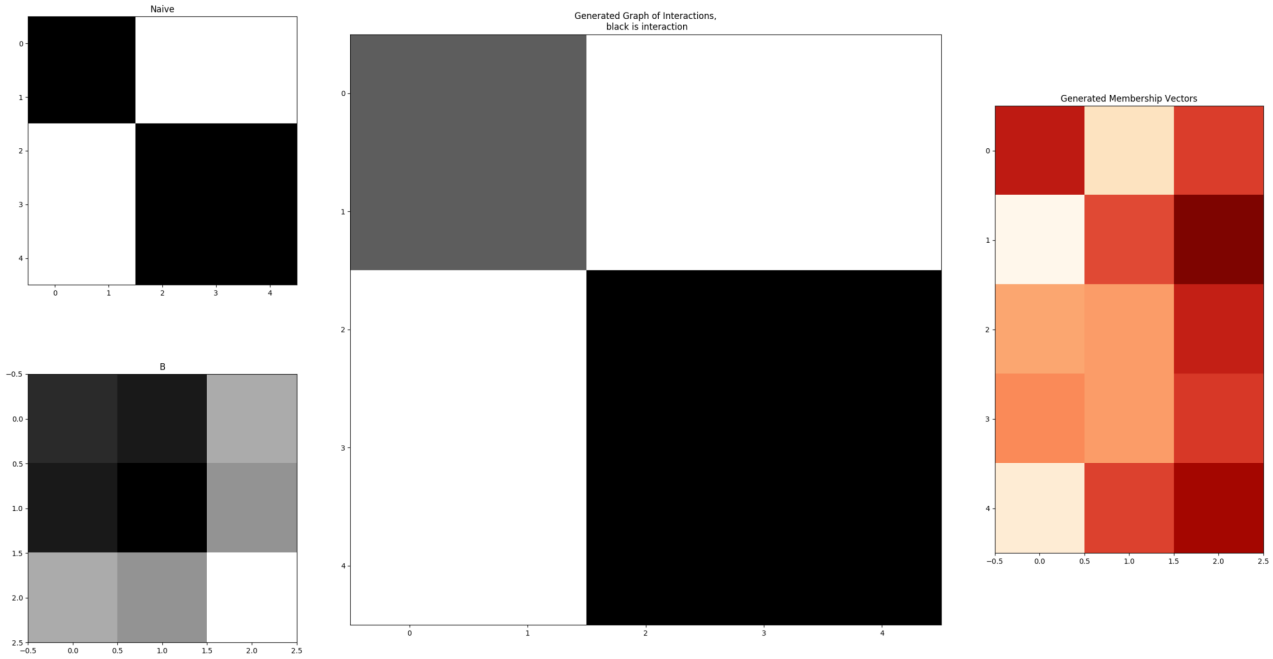


Figure: Results for Naive Inference on 5*5 graph, K=2

# Small Graph

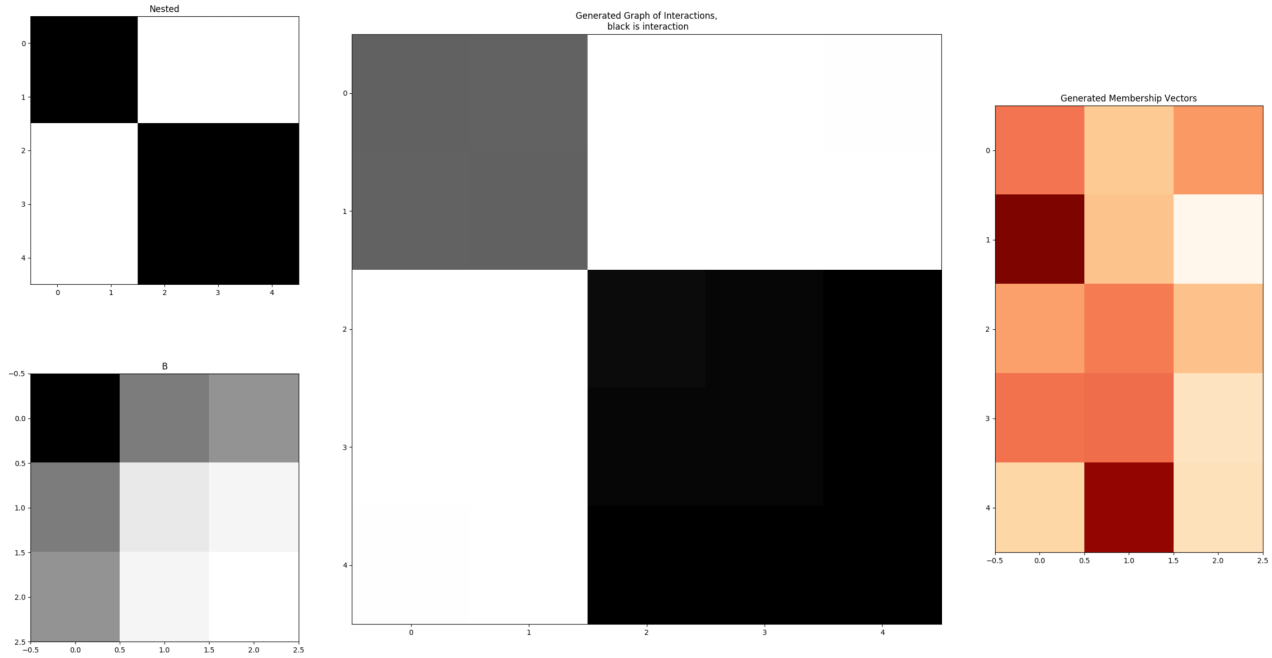


Figure: Results for Nested Inference on 5*5 graph, K=2

# Large Graph

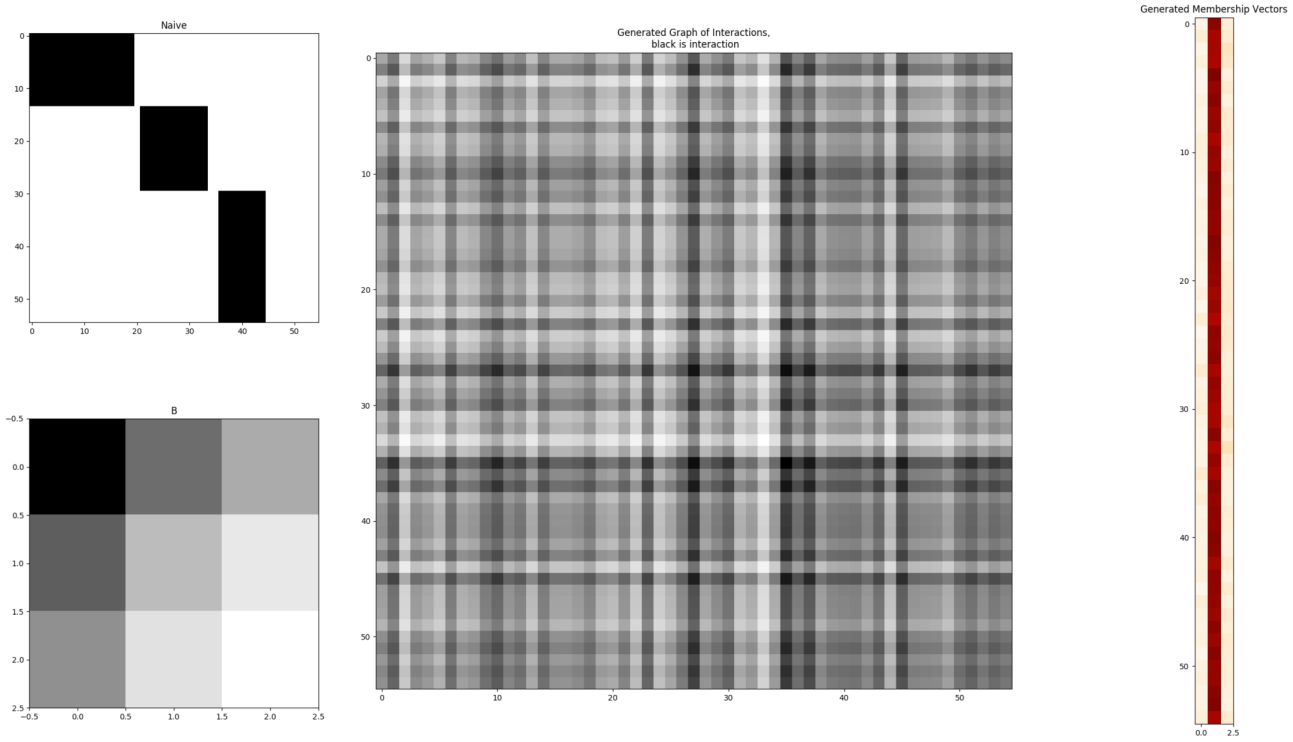


Figure: Results for Naive Inference on 55*55 graph, K=3. Notice poor result

# Large Graph

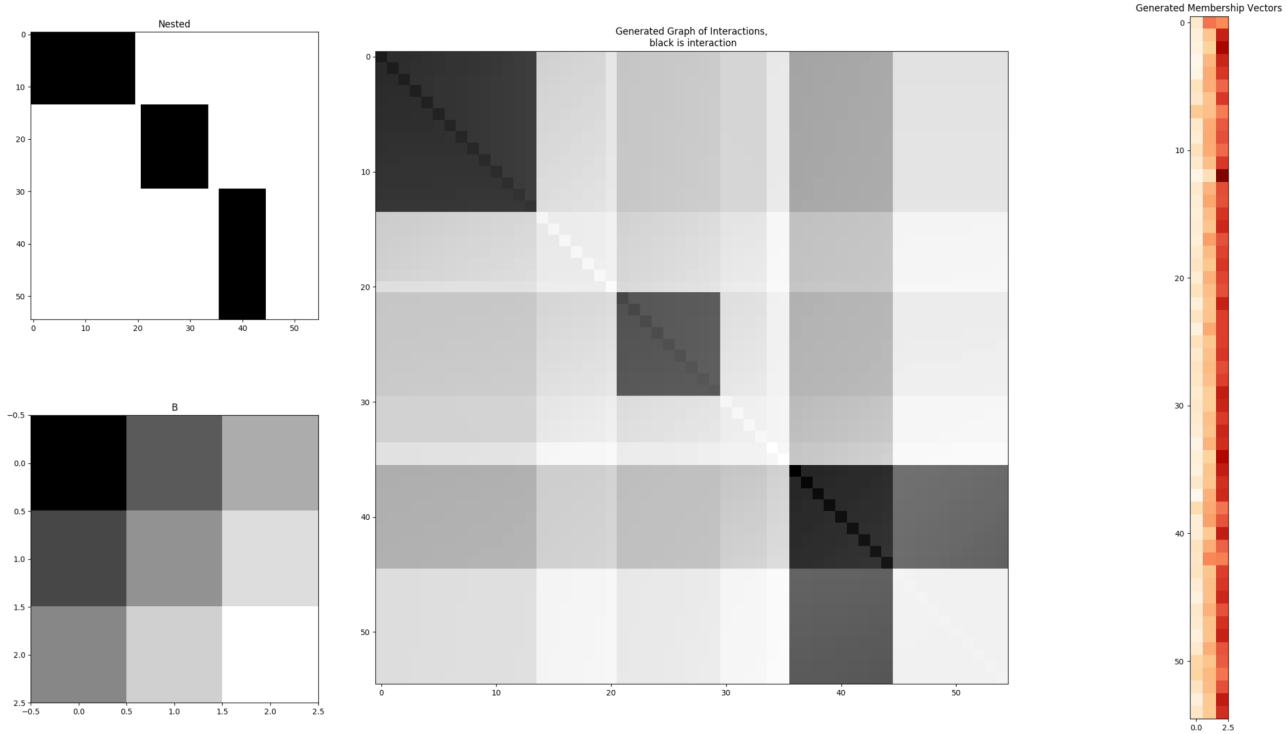


Figure: Results for Nested Inference on 55*55 graph, K=3. Notice that this has converged!

# Learnings and Insights

- MMSB is a complex model to perform inference on, especially when dealing with large, real world networks.
- The variational inference methods discussed above, prove to act as a powerful tool in this context; to infer the large number of variational parameters in this specific model.
- We also noted that scaling-up the method requires utilizing sampling methods and working with subgraphs.
- Link sampling showed that sampling methods that exploit the structure in the problem give better results.
- Finally, we feel that applying techniques such as BBVI to this problem (that reduce the variance of noisy gradients), should be looked into as part of the future work.

Questions ?