Scaling MMSB to Large Datasets

Akshat Jindal 150075 **Shivam Utreja** 150682 **Pratyush Garg** 15807521

Abstract

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. To better report our results, we implement the baseline approach and a few scalable approaches to MMSB, for a qualitative comparison.

1. Problem Statement

Given real-world relational datasets, like the US patent dataset or the dataset on scientific articles; we wish to identify the underlying community structure i.e., clusters of densely connected nodes. In order to do this, we first propose a probabilistic generative story for this kind of data, followed by proposing methods to learn the parameters of this model efficiently.

2. Previous Methods

A plethora of work on relational data existed even before the MMSB model was proposed by Airoldi et al.(1). As we saw in class, latent stochastic blockmodels are also used to model relational data.

2.1 Stochastic Blockmodels

The stochastic blockmodel is an extension/adaptation of mixture models. In that model, each object(data point $\mathbf{x}_n$, belongs to a cluster and the relationships between objects are governed by the corresponding pair of clusters. With posterior inference, one identifies a set of latent roles which govern the objects relationships with each other. A recent extension of this model relaxed the finite-cardinality assumption on the latent clusters with a non-parametric hierarchical Bayesian model based on the Dirichlet process prior.

2.2 Limitations

This model is however a bit too simple for many scenarios as each object can only belong to one cluster, or in other words, play a single latent role. However, many relational data sets are multi-facet. For example, when a protein or a social actor interacts with different partners, different functional or social contexts may apply and thus the protein or the actor may be acting according to different latent roles they can possible play.

Thus came along MMSB where the model associates each unit of observation with multiple clusters rather than a single cluster, via a membership probability-like vector ($\pi_n \ \forall \ n \ \in \mathcal{N}$)

3. The MMSB Model

In this section, we'll describe the MMSB model, disucss the generative story and the naive inference procedure used to infer the posteriors in the model.

The Mixed Membership Stochastic Blockmodel posits a graph $\mathcal{G} = (\mathcal{N}, \mathbf{Y})$ where $\mathcal{N}$ is the number of vertices in the graph and $\mathbf{Y}$ represents the $\mathcal{N}x\mathcal{N}$ adjacency matrix which is essentially binary in nature. We assume that the vertices essentially belong to K factions modelled by a $Kx1$ mixture vector for every vertex and context dependent latent vectors for each vertex. The probabilities of interactions between different factions are defined by a matrix of Bernoulli rates : $\mathcal{B}$

3.1 The Generative Story

1. Draw a K dimensional mixed membership vector $\boldsymbol{\pi}_p \sim$ Dirichlet(α) $\ \forall p \ \in \mathcal{N}$
2. For each pair of vertices $(p, q) \in \mathcal{G}$:
    - Draw membership indicator for the initiator : $z_{p->q} \sim$ Multinomial($\boldsymbol{\pi}_p$)
    - Draw membership indicator for the initiator : $z_{q->p} \sim$ Multinomial($\boldsymbol{\pi}_q$)
    - Sample value of interaction : $Y(p, q) \sim$ Bernoulli($B_{z_{p->q}, z_{q->p}}$)

3.2 Sparsity

Adjacency matrices encoding binary pairwise measurements are often sparse, that is, they contain many zeros or non-interactions. 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.

While the 2^{nd} form of sparsity is captured by MMSB as it infers faction-faction interation strength $\mathbf{B}$, the 1^{st} form is not. A good estimate of the portion of zeros that should not be explained by the blockmodel $\mathbf{B}$ reduces the bias of the estimates of its elements.

Thus, a sparsity parameter $\rho \in [0, 1]$ is introduced in the MMSB to characterize the source of non-interaction. Instead of sampling a relation $Y(p, q)$ directly, we down-weight the probability of successful interaction by $(1 - \rho)$ where the weight ρ captures the portion of zeros that should not be explained by the blockmodel $\mathbf{B}$. A large value of ρ will cause the interactions in the matrix to be weighted more than non-interactions.

3.3 Inference

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

Posterior Inference : The E-Step

So as we know, we'll maximise $\mathcal{L}(q, \Theta^{old})$ in the E-Step wrt q, which gives q as the joint CP of $\boldsymbol{\pi}_p$s and $z_{p->q}$s. Since, this is intractable to calculate, we use Mean-Field VB to get q^{opt} Since we're using a mean-field VB,

we define the variational distribution we take :

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

where q_1 is Dirichlet and q_2 is Multinomial. Thus the variational parameters to be inferred : $\{\boldsymbol{\gamma}, \boldsymbol{\phi}\}$

Solving via the Mean-Field approach taught in class, the update equations for the parameters(They minimize the KL divergence between the variational distribution and the joint distribution of Y and the variational parameters :

$$\phi^{new}_{p->q,g} \propto e^{\mathbf{E}_q[log\pi_{p,g}]} \prod_h ((\mathbf{B}(g,h))^{Y(p,q)}(1-\mathbf{B}(g.h))^{1-Y(p,q)})^{\phi^{old}_{q->p,h}}$$
$$\phi^{new}_{q->p,h} \propto e^{\mathbf{E}_q[log\pi_{q,h}]} \prod_g ((\mathbf{B}(g,h))^{Y(p,q)}(1-\mathbf{B}(g.h))^{1-Y(p,q)})^{\phi^{old}_{p->q,g}}$$
$$\gamma^{new}_{p,k} = \boldsymbol{\alpha}_k + \sum_q \phi^{new}_{p->q,k} + \sum_q \phi^{new}_{q->p,k}$$

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

Point Estimates : The M-Step

Here, using the variational distribution, we maximise the $\mathcal{L}(q^{optimal}, \Theta)$ wrt $\Theta = \{\alpha, \mathbf{B}\}$

Isolating terms containing $\boldsymbol{\alpha}$ we obtain $L_{\boldsymbol{\alpha}}(q^{opt}, \Theta)$. Unfortunately, a closed form solution for the approximate maximum likelihood estimate of $\boldsymbol{\alpha}$ does not exist. We can produce a Newton-Raphson method that is linear in time, where the gradient and Hessian are :

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_k} &= N\left(\psi\left(\sum_k \alpha_k\right) - \psi(\alpha_k)\right) + \sum_p \left(\psi(\gamma_{p,k}) - \psi\left(\sum_k \gamma_{p,k}\right)\right), \\
\frac{\partial \mathcal{L}_{\vec{\alpha}}}{\partial \alpha_{k_1} \alpha_{k_2}} &= N\left(\mathbb{I}_{(k_1=k_2)} \cdot \psi'(\alpha_{k_1}) - \psi'\left(\sum_k \alpha_k\right)\right).
\end{aligned}$$

where $\boldsymbol{\psi}(x)$ is the derivative of the log-gamma function, $\frac{dlog\tau(x)}{dx}$

Isolating terms containing $\mathbf{B}$ we obtain $\mathcal{L}_B$, whose approximate maximum is :

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

However, the authors for experimental purposes, do not learn α, but instead experiment with various values of α because as α increases, each node is likely to belong to more clusters.

The sparsity parameter ρ is also estimated as $1-\hat{d}$, where $\hat{d}$ is the density of the adjacency matrix $= \frac{\sum_{p,q} Y(p,q)}{N^2}$

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.

A discussion

The model decouples the observed connectivity patterns into two sources of variability, B, $\boldsymbol{\pi}$s,that are apparently in competition for explaining the data, possibly raising an identifiability issue. This is not the case, however, as the blockmodel B captures global/asymmetric relations, while the mixed membership vectors s capture local/symmetric relations. This difference practically eliminates the issue, unless there is no signal in the data to begin with.

Reconstruction

To reconstruct our matrix once inference is done, there are two ways of computing posterior model-based expectations of each interaction as follows:

$$\mathbb{E}\left[\, Y(p,q)\,\right] \approx \widehat{\vec{\pi}}_p{}' \widehat{B}\, \widehat{\vec{\pi}}_q \qquad \text{and} \qquad \mathbb{E}\left[\, Y(p,q)\,\right] \approx \widehat{\vec{\phi}}_{p\to q}{}' \widehat{B}\, \widehat{\vec{\phi}}_{p\leftarrow q}.$$

Thus, we can obtain binary interaction networks by thresholding these expected probabilities at different values. This amount to either (i)predicting physical interactions by thresholding the posterior expectations computed using blockmodel B and mixed membership map $\boldsymbol{\pi}$s, essentially a prediction task, or (ii) we de-noise the observed interactions Y using the blockmodel B and interaction specific membership indicators , essentially a de-noising task.

In this sense, between two models that suggest different sets of interactions as reliable, the choice of which one to go for is dependent on us, perhaps taking functional relevance into account as done in gene interaction networks by the authors.

4. Scaling Up

The naive inference procedure for MMSB which the authors call as *naive variational inference* is not scalable. In this algorithm, one initializes the variational Dirichlet parameters $\boldsymbol{\gamma}$ and the variational multinomial parameters $\boldsymbol{\phi}$ to non-informative values, and then iterates the following two steps until convergence:

- Update $\phi_{p->q}$ and $\phi_{q->p}$ for all edges (p, q).
- Update γ_p for all nodes $p \in \mathcal{N}$

The issue is, in such an algorithm, at each variational inference cycle we need to allocate $NK+2N^2K$ scalars.

Another major issue is that the naive algorithm infact shows poor convergence rates. This is because it doesnt capture the dependence between $\boldsymbol{\gamma}$s and $\mathbf{B}$ because at each iteration(t) of the EM algorithm, we conduct the entire VB cycling, we initialise the $\boldsymbol{\gamma}$s to the same non-informative value. This suddenly dampens the likelihood by breaking the dependence between the estimates of parameters in $\boldsymbol{\gamma}^t$s and $\mathbf{B}^t$. This leads to slow convergence!

4.1 Nested VI

Nested VI takes care of this by rescheduling the parameter updates. The $\boldsymbol{\gamma}$s are initialised to non-informative values only once right at the beginning. Thus, we dont start from scratch at every variational cycle. And

the dependence is maintained by always keeping the block of free parameters, $\phi_{p->q}$ and $\phi_{q->p}$, optimized given the other variational parameters.

Furthermore, the nested algorithm has the advantage that it trades time for space thus allowing us to deal with large graphs; at each variational cycle we need to allocate NK + 2K scalars only.

A summary of the algorithm can be seen in the following figure taken from the MMSB paper.

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

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

Figure 5: **Top:** The two-layered variational inference for $(\vec{\gamma}, \phi_{p\to q,g}, \phi_{p\leftarrow q,h})$ and $M = 1$. The inner algorithm consists of Step 5. The function f is described in details in the bottom panel. The partial updates in Step 6 for $\vec{\gamma}$ and B refer to Equation 4 of Section B.4 and Equation 5 of Section B.5, respectively. **Bottom:** Inference for the variational parameters $(\vec{\phi}_{p\to q}, \vec{\phi}_{p\leftarrow q})$ corresponding to the basic observation $Y(p,q)$. This nested algorithm details Step 5 in the top panel. The functions f_1 and f_2 are the updates for $\phi_{p\to q,g}$ and $\phi_{p\leftarrow q,h}$ described in Equations 2 and 3 of Section B.4.

Figure 1: PseudoCode

4.2 Graph Sampling Strategies

Since we are dealing with real-world graphical data, the graphs will have nodes in the order of millions. Hence, any inference algorithm that is tractable on such large data sets will have to be stochastic in nature. Gopalan et al. (2) suggest various sampling techniques, that when combined with Mean-field VI, lead to faster inference algorithms (Stochastic VI) even on very large datasets. They also suggest a couple of well justified, simplifying assumptions that further speed up inference without causing under-fitting.

Simplifying Assumptions

The following two assumptions further simplify the model, beyond what is described in section 3.1.

- *Assortative Undirected Network:*
  This assumption states that two nodes in a given undirected network form a link with each other with high probability, only if they belong to the same community/cluster. In the context of the

MMSB model, this implies that two nodes, i and j form a link with some high probability β_k only if $\phi_{i\to j} = \phi_{i\leftarrow j} = k$. If $\phi_{i\to j} \neq \phi_{i\leftarrow j}$, the probability of link formation is some small, constant value ϵ. Compared to the model described in 3.1, this means that $B_{ij} = \epsilon$ if $i \neq j$ and $B_{kk} = \beta_k, \forall k \in \{1 \dots K\}$.

- *Link, Non-link Distinction:*
  In real-world networks, as the number of nodes increases, the number of links becomes far less than $O(N^2)$. Hence, it becomes increasingly redundant to separately model the interaction parameters for each pair of nodes, regardless of whether a link exists between them. Hence, this assumption aims to overcome this redundancy. It states that we should model only the interaction parameters associated with the links present in the training set. The non-link interaction parameters for a given are assumed to be the mean of the link interaction parameters of that node.

Modified Posterior Updates

Since we are using the mean-field variational inference method, we will have to choose a form of the posterior approximation that we will use in our SVI algorithm. The form of the posterior approximation, however, will depend on which of the two simplifying assumptions given above we use.

- Using assumption 1:

$$q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{\lambda}, \boldsymbol{\gamma}, \boldsymbol{\Phi}) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{i<j} q(z_{i\to j}|\phi_{i\to j}) q(z_{i\leftarrow j}|\phi_{i\leftarrow j}) \tag{1}$$

  Note that this is less expressive when compared to the generative story in section 3.1.

- Using assumptions 1 & 2:

$$q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}) = \prod_{k=1}^{K} q(\beta_k|\lambda_k) \prod_{n=1}^{N} q(\theta_n|\gamma_n) \prod_{(i,j)\in links} q(z_{i\to j}, z_{i\leftarrow j}|\phi_{ij}) \prod_{(i,j)\in nonlinks} q(z_{i\to j}|\phi_{i\to j}) q(z_{i\leftarrow j}|\phi_{i\leftarrow j}) \tag{2}$$

Now looking at the forms of $q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$ above, note that the number of ϕ's are N^2-many, whereas λ's are just K-many and γ's are just N-many. Further note that, λ's and γ's are common for all the links (and non-links) in the network, whereas the ϕ's are link specific, and also depend on the values of λ and γ. Hence, we adopt the notation of calling ϕ's as our local parameters, and λ's and γ's as our global parameters.
Given this notation, the algorithm for the ELBO optimization (The stochastic VI algorithm) of the MMSB is given below:*(Note: This algorithm is with respect to the first form of $q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$ above, which uses only assumption 1.)*

1. Initialize global parameters $\boldsymbol{\gamma} = (\gamma_n)_{n=1}^N$, $\boldsymbol{\lambda} = (\lambda_k)_{k=1}^K$.
2. Subsample a set $\mathcal{S}$ of node pairs.
3. Local step. For each pair $(i,j) \in \mathcal{S}$, compute the optimal interaction parameters $\boldsymbol{\phi}_{i \to j}$ and $\boldsymbol{\phi}_{i \leftarrow j}$ as a function of the global parameters.
4. Global step.
   - For each node a, compute the community membership natural gradients $\partial \gamma_a^t$ and update γ_a.
   - For each community k, compute the community strength natural gradients $\partial \lambda_k^t$ and update λ_k.
5. Repeat.

Figure 2: SVI for MMSB.

Note that in the above equations, the exact form of the updates $\partial \gamma_a^t$, $\partial \lambda_k^t$, $\phi_{i \to j}$ and $\phi_{i \leftarrow j}$ will depend on the sampling scheme used and the form of the posterior approximation $q(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$. These have been discussed in detailed in (2).

Sampling Techniques

We have already highlighted how crucial sampling is in this problem's setting. Note that the posterior updates use the sampling step as a black box. Hence, any sampling technique will work, as long as the noisy gradient is unbiased, and its variance can be controlled. The variance of the noisy gradient can be reduced by increasing the size of the mini-batch used at each iteration. Each of the three sampling techniques discussed below give a noisy, unbiased estimate of the true natural gradient of the ELBO of the posterior.

- *Random Pair Sampling:*
  This is the easiest way of sampling ϕ's. In this method, a small mini-batch from the $N(N-1)/2$ possible node pairs is sampled. For each of the node pairs sampled this way, both $\phi_{i \to j}$ and $\phi_{i \leftarrow j}$ are used and updated in the current AltOpt iteration. Note that this does, indeed, give an unbiased estimate of the natural gradient because the ϕ's are being sampled uniformly.

- *Random Node Sampling:*
  At each iteration, one of the N nodes is uniformly sampled. Then we consider all of the ϕ's associated with the sampled node, irrespective of whether a link exists or not. For instance, if node i is sampled in a given iteration, we will use all the $\{\phi_{i \to j}\}$ and $\{\phi_{i \leftarrow j}\}$, for all $j \neq i$; for the updates in the current iteration.

- *Link Sampling:*
  This sampling technique makes use of the second simplifying assumption (and its corresponding posterior approximation) mentioned above. In this method, we first sample a mini-batch of nodes uniformly (without repetition). Now for each of the sampled nodes $\{i\}$, we use all the $\{\phi_{i \to j}\}$'s and $\{\phi_{i \leftarrow j}\}$'s for all $(i,j) \in links(i)$; in the current iteration of the SVI update. The ϕ's associated with the non-links for a specific node i are updated using the fact that they are constrained to be the mean of the interaction parameters associated with the links of the node i. As shown in (2), the relationship between the non-link and the link ϕ's is shown below. Here d_i is the degree of node i in the data.

$$\phi_{i \to m,k} = \frac{\Sigma_{(i,j) \in links(i)} \phi_{ij}^{kk}}{d_i} \tag{3}$$

The key advantage of this sampling scheme is that we can use *all* the links of the sampled node from the training set (because they are already sparse). We can hence, use the noise-free, true natural gradients when updating the community membership parameters for the links.

Initialization of Parameters

The algorithm heavily relies on initialization of the number of communities; and convergence rate also depends on the initial values of the community membership vectors for every node. While a random intialization is possible, (2) also highlights a better alternative algorithm that exploits the assumption that every link is indicative of common community membership (which is a logical *starting* point).

The algorithm begins by calculating random membership vectors for each node and then assigning each node to its own community by adding a positive weight to the $\gamma_{n,n}$. Then, for every link (a,b) in the training set, the algorithm adds weight in a for the community dominant in b (till the previous iteration) and vice versa. This amounts to an exchange in the dominant communties for both the nodes. This is run for a total of log N epochs (where N is the number of nodes), under the 'small world' assumption.

When the algorithm terminates, we have a list of communities where a node belongs to community k if it is within a single step of another node, that has a high ($\geq$ threshold) probability of belonging to k. These lists can be used for initialization of the membership vectors by adding greater weights to the communities so predicted for each node. The algorithm, as formulated in (2), is given in Figure 2.

1. Initialize variational parameters of MMSB model M.
   - M has N nodes and N communities.
   - Initialize $\gamma = (\gamma_n)_{n=1}^{N}$ randomly.
   - Assign each node n to its own community n by adding a small positive weight to $\gamma_{n,n}$.
   - Keep only the top r communities of each node.
2. For each link (a,b) in the training set,
   - Let t_a and t_b be the top communities of nodes a and b.
   - Set $\gamma_{a,t_b} \leftarrow \gamma_{a,t_b} + 1$; $\gamma_{b,t_a} \leftarrow \gamma_{b,t_a} + 1$.
3. Recompute the top r communities of each node.
4. Repeat steps 2, 3 for log N iterations.
5. For each link (a,b) in the training set,
   - Assign nodes a and b to community k if the approximate posterior probability $p(z_{a\to b} = z_{a\leftarrow b} = k|\mathbf{y}, M) > 0.5$.
6. Return the overlapping communities and their cardinality.

Figure 3: Initialization Algorithm

Computational Complexity

As described above, the algorithm is divided into the global and the local step, updates of which take separate times. Since, we take the assortative assumption, the complexity for the local VI step is not quadratic in K but instead O(SK) where S is the number of sampled links and K is the number of communities. The global step is of the order O(NK), where N is the number of nodes, if we update the membership vectors for each after every iteration. Note that we can also scale this up by using a minibatch.

For eg, the link sampling technique for SVI has a complexity of the order O(MK+nK), with M as number of links and n as minibatch size.

Experimental Verification

The authors of (2) provided code for the implementation of the model, however, it did not age well with changing libraries and compilers, making our testing very difficult. We did, though, find results for a toy problem similar to the one we implement in the baseline and nested VI MMSB. (Section 5). The following is an example file from the original implementation by Gopalan and Blei:

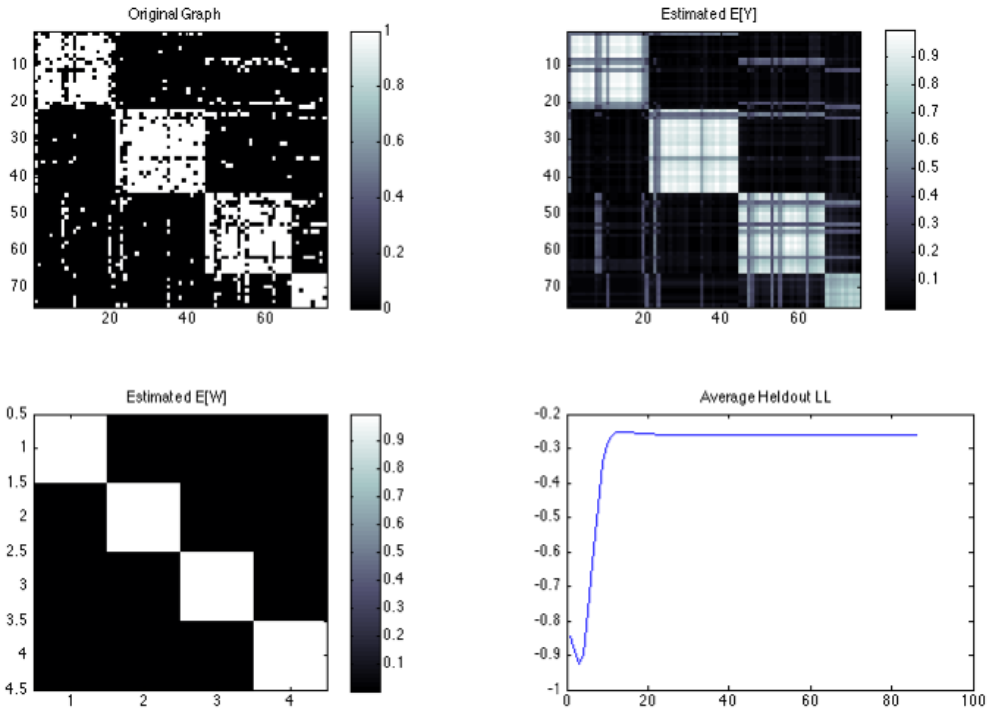

Figure 4: SVINET on a Toy Problem: N=75, K=4

For results of overlapping communities on real world networks like the US patents database and the arXiv database, we refer you to the paper itself.

5. Implementation attempted: Baseline vs Nested VI

We implemented the naive inference algorithm as proposed by the authors in the paper which uses naive Variational EM.

We then implemented the nested VI algorithm as described in the paper so that we could compare and experimentally check how good nested VI actually is.

Small Graph

We first ran both the inference algorithms for a small toy graph with and adjacency matrix Y of dimension $5 * 5$ and designed the graph to have 2 latent factions. Thus setting K=2 ($\alpha = 0.5$), we ran the algorithms.

Both the algorithms converged in nearly the same amount of time and the results(where we basically show a reconstruction of the adjacency matrices) clearly discover the latent communities. Below we show the results for both the algorithms. The top left box shows the true adjacency matrix, while the large central box is the reconstructed version. We also show $\mathbf{B}$ in the bottom left and the membership vectors in red for each algorithm.

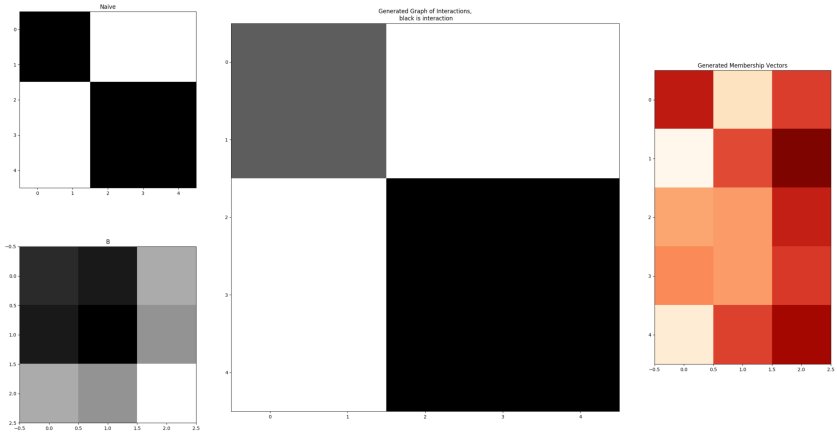

Figure 5: Results for Naive Inference on 5*5 graph, K=2

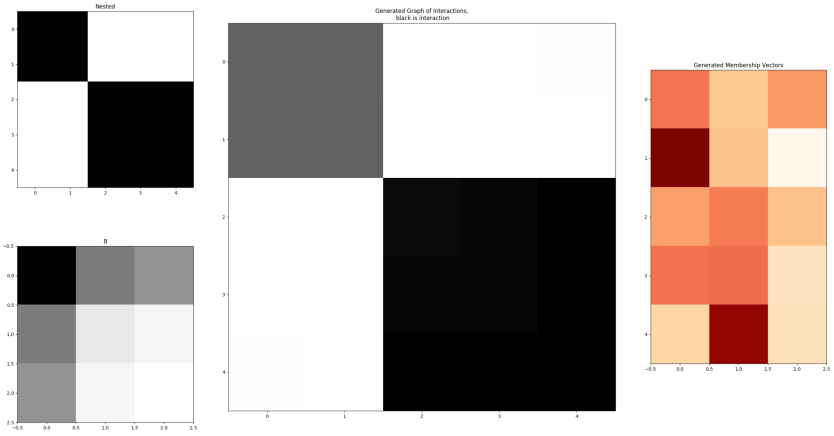

Figure 6: Results for Nested Inference on 5*5 graph, K=2

Large Graph

We then ran the algorithms on a larger toy graph to check if nested VI converged faster. The graph's adjacency matrix Y was of dimension $55 * 55$ and was designed to have 3 latent factions. Thus setting K=3 ($\alpha = 0.5$), we ran the algorithms.

Here we saw that in the amount of time that the Nested VI algorithm converged and succesfully discovered the 3 latent communities, the naive inference procedure still hadn't converged well enough and produced poor result. Below we show the results when both algorithms were run for the same time `t : The convergence time for Nested V`

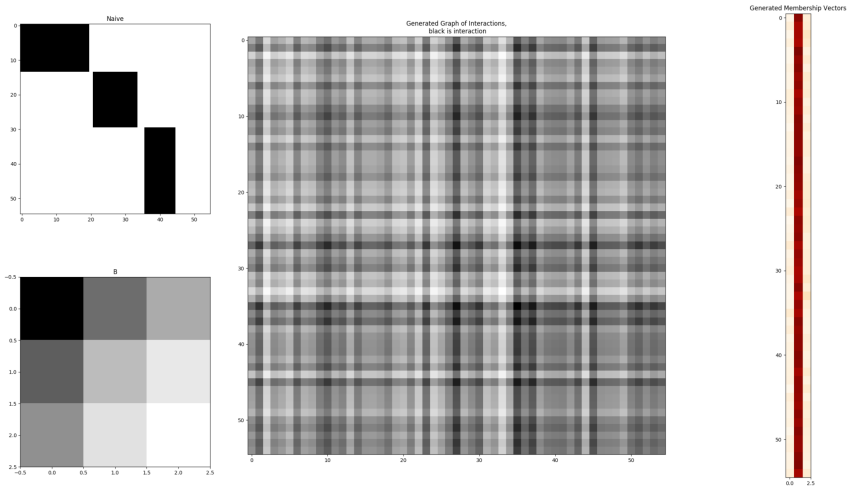

Figure 7: Results for Naive Inference on 55*55 graph, K=3. Notice poor result

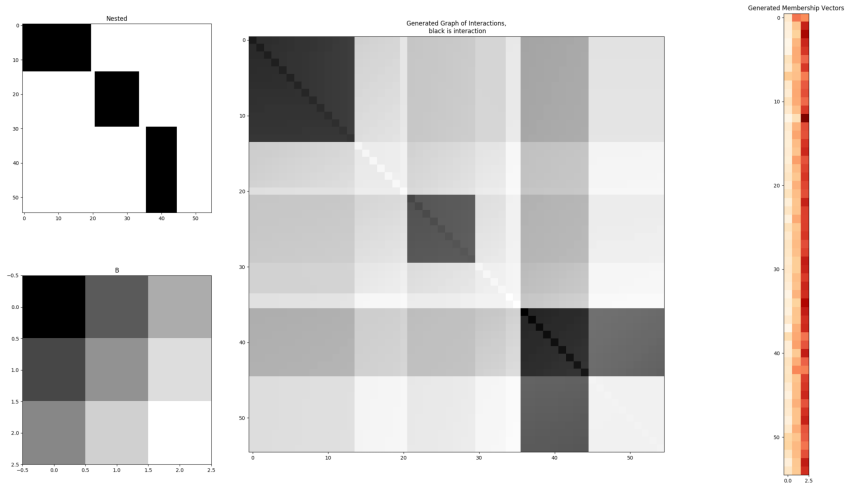

Figure 8: Results for Nested Inference on 55*55 graph, K=3. Notice that this has converged!

6. Insights and Possibilities for Future Work

1. MMSB is a complex model to perform inference on, especially when dealing with the problem of scaling to large, real world networks. Most of the work in furthering MMSB is in trying to reduce its complexity and sensitivity to the size of the network.

2. 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. Both, the nested VI method and the sampling methods provide useful ways of scaling for space and time respectively.

3. When comparing the two, we find that though nested VI in (1) gives a more flexible model for inference, it can be outperformed on real world networks that follow the assumptions of the subsampled SVI.

4. Link sampling showed that sampling methods that exploit the structure in the problem give better results and therefore opens the door for similar smarter methods for sampling.

5. Finally, we feel that applying newly popularised techniques such as BBVI to this problem (that reduce the variance of noisy gradients), should be looked into as part of the future work.

References

[1] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1390681.1442798`.

[2] Prem K. Gopalan and David M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013. ISSN 0027-8424. doi: 10.1073/pnas.1221839110. URL `https://www.pnas.org/content/110/36/14534`.