Dynamic Clustering for Network Data

Xuecong Fu (xuecongf) [1] **Wanshan Li (wanshanl)** [1] **Anusha Sinha (anushasi)** [1]

1. Introduction and related work

In recent years, there has been increasing interest in studying random graphs and network models. This is because network or relational data occurs more and more frequently in many research fields, such as the social sciences, biological sciences, and civil engineering. Furthermore, studying the evolution of the networks and relationships induced by the the data over time has the potential to reveal more information of the structure underlying the data. In certain applications, the evolution of the network over time can be the primary motivation for the research.

In the field of network modeling, there has been much interest on latent variable models due to the fact that introducing latent variables make the dependence between edges more explicitly by network analysis. For example, latent variables allow for the computation of conditional independence in error terms (Kim et al., 2018). This area of research has focused specifically on two classes of models in recent years, latent space models (LSM) and stochastic blockmodels (SBM). Latent space models assume that if nodes are closer in latent space, there tend to exist edges between them, so they allow for useful visualization and interpretation of network data because there tend to exist more intra-cluster edges than inter-cluster edges (Hoff et al., 2002). Stochastic blockmodels, on the other hand, assume that the nodes in the network are divided into different hidden classes and various frameworks of analysis exist where the latent labels of the nodes are either known or unknown.

There are many well-known challenges in modeling dynamic networks. Computational complexity is one of them, and it is what we would focus on in our project. Direct inference, even with simple model setting, is an NP-Hard problem for stochastic blockmodels. Hence without smart inference techniques the model parameters can be infeasible to calculate for networks larger than a few thousand nodes (Rastelli et al., 2016). We will try to implement several techniques in the literature for approximate and fast inference, like variational inference and Kalman filter, in our project, and try to make our own improvement over existing methods.

[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

1.1. Clustering in Network Data

Assume that $G = \{V, X\}$ is a network or a random graph, where V is the set of nodes and $|V| = n$, and $X = \{X_{ij}\}_{i,j=1}^{n}$ is the set of edges. By its name, X_{ij}'s are random variables. In network data, what we observe is a set of $\{G\}$ random graphs. Latent space model (Hoff et al., 2002) is a powerful tool for modeling network data, which refers to the general form of a series of models. It assumes that the distribution of X_{ij} depends on some unobserved variables called *latent variables*. A (simple form of) latent space model assumes

$$P(X_{ij} = x | Z_i, Z_j, Y_{ij}) = f(x; Z_i, Z_j, Y_{ij}, \theta) \quad (1)$$

for some parameters θ, and Z_i, Z_j, Y_{ij} are the latent variables of node i, j and their edge. For our particular interest in this project, the clustering of nodes in a network, a possible way is to cluster the latent variables Z_i's of nodes in the latent metric space.

Apart from this general approach, there is a special form of latent space model called stochastic block model (Holland et al., 1983) which is specifically tailored for cluster structures in network data. The usual stochastic block model assumes that the n nodes belongs to K clusters or groups, with the cluster label of node i denoted by C_i, and the distribution of X_{ij} is Bernoulli, given by

$$P(X_{ij} = 1 | C_i, C_j) = \beta_{C_i, C_j}, \quad i, j = 1, \cdots n \quad (2)$$

Compactly, if we let π_i be the one-hot representation of the cluster label of node i, i.e., π_i is binary and $\pi_i(j) = 1$ if and only if $j = C_i$, and let $\Pi = [\pi_1, \cdots, \pi_n]^T \in \mathbb{R}^{n \times K}$, $B = [\beta_{kl}]_{k,l=1}^{K}$, then

$$E(X|\Pi) = \Pi B \Pi^T. \quad (3)$$

Stochastic block model is frequently used in clustering in network data, because the *connectivity matrix* $B = [\beta_{kl}]_{k,l=1}^{K}$ naturally describes the cluster structure: usually the diagonal of B is larger than off-diagonals, meaning that nodes within a cluster are connected to each other more likely than nodes between different clusters.

There is an important generalization of the original SBM, called Mixed-membership Stochastic Blockmodel (MMSB) (Airoldi et al., 2009). It allows the *membership vector* π_i to be a non-binary probability mass vector, so that it can model the probability of node i belongs to the K clusters, i.e., $P(C_i = k) = \pi_i(k)$. In this case the matrix form (3) still holds. It is more useful in modeling real-world data, as in many applications we need soft-clustering where a node can belong to more than one clusters, rather than hard-clustering.

1.2. Dynamic Clustering

There are several different approaches to modeling dynamics of networks, including but not limited to: by assuming some underlying stochastic process (Perry & Wolfe, 2013), by spectral clustering (Liu et al., 2018; Pensky & Zhang, 2019), and by using state-space model (Sarkar & Moore, 2006; Xing et al., 2010; Kolar et al., 2010; Xu & Hero, 2014; Cho et al., 2014). For the purpose of this project, we are particularly interested in the state-space model approach.

Essentially, a state-space model is a probabilistic graphical model that describes the probabilistic dependence between the latent state variable and the observation. The hidden Markov-chain model is a special case of state-space model. To the best of our knowledge, all papers on dynamic network modeling using state-space model approach actually assumes an HMM structure. By Figure 1, a dynamic network model needs assumptions on the transition distribution $P(Z^{(t)}|Z^{(t-1)})$ and the generation distribution $P(X^{(t)}|Z^{(t)})$.

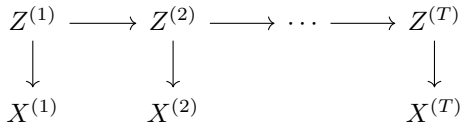


Figure 1. A hidden Markov model.

The earliest paper considering dynamic network modeling is (Sarkar & Moore, 2006). They assume that $Z_i^{(t)} \in \mathbb{R}^n$ and $P(X_{ij}^{(t)} = 1|Z_i^{(t)}, Z_j^{(t)}) = f(\|Z_i^{(t)} - Z_j^{(t)}\|_2)$, and $\log P(Z^{(t)}|Z^{(t-1)}) = \frac{1}{2\sigma^2}\|Z^{(t)} - Z^{(t-1)}\|_2^2 + c$ which some function f and constant c. When solving the HMM, they didn't use variational inference as usual, instead they use derive the close-form gradient of the log-likelihood function and use warm-start and a special kernel function to reduce the computation price. For the warm-start solution, they optimize an objective function which is essentially simultaneous MDS at T time points with penalization on the time-variation:

$$\min \left|\tilde{D}_t - X^{(t)}X^{(t)\top}\right|_F + \lambda \left|X^{(t)}X^{(t)\top} - X^{(t-1)}X^{(t-1)\top}\right|_F.$$

Interestingly, (Liu et al., 2018) uses the same trick, i.e., penalization for dynamic clustering.

(Xing et al., 2010; Xu & Hero, 2014; Cho et al., 2014) are three representatives of dynamic generalization of (mixed-membership) stochastic block models. For SBM and MMSB, we need to model the following two transition distributions:

$$P(\Pi^{(t)}|\Pi^{(t-1)}), P(B^{(t)}|B^{(t-1)}).$$

One should notice that the complexity of network data with time-stamps can make it infeasible to fit very simple model. To use variational inference technique, we need smart designs of the transition distributions so that an appropriate approximation can be figured out.

Normal distribution is relatively easy to handle, so we can use normal distribution for transition, and use some deterministic transformation to keep the hyperparameters in a specific range. This is the trick in (Xing et al., 2010). They use a logistic normal distribution to model $\{\pi_i\}$, and use a transition matrix A to model $\mu^{(t)}|\mu^{(t-1)} \sim \text{Normal}(A\mu^{(t-1)}, \Phi)$ where $\mu^{(t)}$ is the mean vector of the logistic normal. For entries $\beta_{kl}^{(t)}$ of $B^{(t)}$, they assume $\beta_{kl}^{(t)} = \text{logit}(\eta_{kl}^{(t)})$ and $\eta_{kl}^{(t)}|\eta_{kl}^{(t-1)} \sim \text{Normal}(b\eta_{kl}^{(t-1)}, \psi)$. Even under these assumptions, the inference for the whole model is very complex. Implementing their algorithms would be the start of our project. (Cho et al., 2014) use similar settings for transition distributions. Both papers eventually assume that $B^{(t)} = B$ for all t to simplify the inference algorithm.

On the other hand, (Xu & Hero, 2014) uses other techniques like label-switching algorithm and extended Kalman Filter to handle time-varying connectivity matrix $B^{(t)}$. A time-varying $B^{(t)}$ can definitely reveal more information in the data.

2. Ideas and Methods

2.1. Model: Dynamic LNMMSB

For this project, we implemented a modified version of the mixed membership stochastic blockmodel, specifically the dynamic logistic-normal mixed membership stochastic blockmodel proposed in (Xing et al., 2010). The goal of this paper was to develop a model to infer network tomography via posterior inference from dynamically evolving networks $\{G^{(1)}, \ldots, G^{(T)}\}$ where $G^{(t)} = \{V, E^{(t)}\}$ and it is assumed that $N = |V|$ remains constant over time. We also assume that each vertex can take on multiple roles or function, where the proportion of the involved

roles/functions can be represented by a vector $\vec{\pi}_i^{(t)} \in \mathbb{R}^K$. Links between vertices are determined stochastically according to a compatibility matrix defined over all possible pairs of roles, $B = \{\beta_{k,l}\}$. In (Xing et al., 2010), they assume that the role-compatibility matrix evolves over time, but we implemented a simpler version of the model that assumes B does not vary over time. We planned to first use the provided framework to implement a static logistic-normal mixed membership stochastic blockmodel (LNMMSB), and then implement the expanded version of this framework for the dynamic network case (dMMSB).

For the static framework (LNMMSB), each vertex has a mixed membership vector $\vec{\pi}_i = [\pi_{i,1}, \ldots, \pi_{i,K}]$, which we assume to follow a log normal distribution, where $\vec{\gamma}_i \sim \mathcal{N}(\vec{\mu}, \Sigma)$ and $\pi_{i,k} = \exp\{\gamma_{i,k} - C(\vec{\gamma}_i)\}$ where $C(\vec{\gamma}_i) = \log\left(\sum_{k=1}^K \exp\{\gamma_{i,k}\}\right)$. In the dynamic framework, we define a state-space model for each mixed membership prior and a logistic-normal mixed membership model for networks at each time step. We assume that $\vec{\mu}$ is time-varying such that $\vec{\mu}^{(t)} \sim \text{Normal}(\mathbf{A}\vec{\mu}^{(t-1)}, \Phi)$ and $\vec{\mu}^{(1)} \sim \text{Normal}(\nu, \Phi)$. Then for each node of the network at each timestep, we assume $\vec{\pi}_i^{(t)} \sim \text{LogisticNormal}(\vec{\mu}^{(t)}, \Sigma^{(t)})$. For each pair of nodes $(i,j) \in [1,N] \times [1,N]$, we assume the membership indicator for the donor, $\vec{z}_{i\to j}^{(t)} \sim \text{Multinomial}(\vec{\pi}_i^{(t)}, 1)$ and the membership indicator for the acceptor, $\vec{z}_{j\leftarrow i}^{(t)} \sim \text{Multinomial}(\vec{\pi}_j^{(t)}, 1)$. The links between nodes $e_{i,j}^{(t)} \in E^{(t)}$ are assumed to follow $e_{i,j}^{(t)} \sim \text{Bernoulli}(\vec{z}_{i\to j}^{(t)'} B \vec{z}_{j\leftarrow i}^{(t)} \sim \text{Multinomial}(\vec{\pi}_j^{(t)}, 1))$.

2.2. Variational EM + RTS smoother

Direct EM estimation of the model parameters ν, $\vec{\mu}^{(t)}$, B, and $\Sigma^{(t)}$ proves to be infeasible because of difficulties in marginalization over $\vec{\mathbf{z}}$ and $\vec{\pi}$ and integration of $\vec{\pi}$ under a logistic normal prior (Xing et al., 2010). Instead, we implement a Laplace variational approximation scheme, following algorithms presented in (Xing et al., 2010).

By using generalized mean field, we can approximate the joint posterior with the product of several independent marginal probabilities of desired probabilities $q(\vec{\gamma}), q(\vec{z}_{i\to j}^{(t)}), q(\vec{z}_{j\leftarrow i}^{(t)})$. The variational EM would iteratively update the variational parameters and the model parameters. The updated equations for

$$q_z(\vec{z}_{i\to j}^{(t)}, \vec{z}_{i\leftarrow j}^{(t)}) \sim \textbf{Multinomial}(\delta_{\vec{ij}})$$

, where $\delta_{ij(u,v)} = \frac{1}{C}\exp(\langle\gamma_{i,u}\rangle_{q_\gamma} + \langle\gamma_{j,v}\rangle_{q_\gamma})\beta_{u,v}^{e_{ij}}(1-\beta_{u,v})^{1-e_{ij}}$.

$$\vec{z}_{i\to j}^{(t)} = \sum_v \delta_{ij(u,v)}$$

$$\vec{z}_{j\leftarrow i}^{(t)} = \sum_u \delta_{ij(u,v)}$$

.

For the equation for $q_\gamma(\vec{\gamma}_i)$, it is unintegratable thus impossible for a close form for update. Therefore, another approximation is applied via a Taylor expansion, which turns it into a multivariate normal distribution with

$$\tilde{\Sigma} = (\Sigma^{-1} + (2N-2)H)^{-1}$$

$$\tilde{\gamma} = \vec{\mu} + \tilde{\Sigma}(\langle\vec{m_i}\rangle_{q_z} - (2N-2)\vec{g} + (2N-2)H\hat{\gamma_i} - (2N-2)H\vec{\mu})$$

, where

$$g(\hat{\gamma}) = \frac{\exp\hat{\gamma}}{\sum\exp\hat{\gamma}}$$

$$H = diag(\vec{g} - \vec{g}\vec{g}^T)$$

After our experiments, we found it useful to do multiple initialization for $\tilde{\gamma_i}$ and pick the one with the best likelihood.

In our implementation, we let $\mathbf{A}$ be constant and define it to be the identity matrix and define $\Phi = \sigma I$ where σ is a constant. This reduces the model to a random walk in the membership-mixing space and implies that the mixed-membership vectors for each node may vary widely over adjacent timesteps. (Xing et al., 2010) notes that this model is not ideal, but that even a random walk should provide a better fit of the data than a static model that ignores the time stamps of the networks and considers them individually.

2.2.1. Parameter Estimation for Logistic-Normal MMSB

For this implementation, we ignore the time evolution of $\vec{\mu}$ and estimate model parameters using a straightforward EM-style algorithm. In the E-step, we compute the posterior distribution and expectation of the latent variables by fixing the current parameters. That is, we update $q(\vec{z}_{i\to j}, \vec{z}_{j\leftarrow i}) \sim \text{Multinomial}(\vec{\delta}_{ij})$ and $q(\vec{\gamma}_i) \sim \text{Normal}(\tilde{\gamma}_i, \tilde{\Sigma}_i)$. In the M-step, we update the model parameters by maximizing the log-likelihood obtained from the E-step using the update formulas for variational EM provided in (Xing et al., 2010):

$$\hat{\beta}_{k,l} = \frac{\sum_{i,j} e_{ij}\delta_{ij(k,l)}}{\sum_{i,j}\delta_{ij(k,l)}},$$

$$\hat{\Sigma} = \frac{1}{N}\sum_i \tilde{\Sigma}_i + \text{Cov}(\tilde{\gamma}_{1:N}),$$

$$\hat{\mu} = \frac{1}{N}\sum_i \tilde{\gamma}_i$$

This algorithm is repeated until the relative change of log-likelihood is less than 10^{-6} in absolute value.

2.2.2. Parameter Estimation for dMMSB

Estimation of $\vec{\mu}^{(t)}$ must be done under an SSM, and $\vec{\mu}$ that appeared in the equations above for LNMMSB must be replaced by the posterior mean of $\vec{\mu}^{(t)}$ under this SSM. Given the estimates $\{\tilde{\gamma}_i^{(t)}\}$ returned by the LNMMSB at each time-point, posterior inference on $\vec{\mu}^{(t)}$ can be done using the Kalman Filter (KF) and Rauch-Tung-Streibel (RTS) smoother algorithms (Xing et al., 2010).

Inference for dMMSB can therefore be summarized as

1. Initialize all $\vec{\mu}^{(t)}$
2. While not converged
   (a) For each t
      i. Call the inference algorithm for LNMMSB on network $E^{(t)}$ and return the approximation $\tilde{\gamma}_i^{(t)}$, $\tilde{\Sigma}_i^{(t)}$
      ii. Update the observations $\vec{Y}^{(t)} = \sum_i \tilde{\gamma}_i^{(t)}/N$
   (b) KF and RTS smoother update $\vec{\mu}^{(t)} = \hat{\mu}_{t|T}$ based on $\{\vec{Y}^{(t)}\}_{t=1}^T$

The full variational EM algorithm for dMMSB consists of an E-step and M-step as above for LNMMSB. In the E-Step, the above inference algorithm is used to compute $\hat{\mu}_{t|T}$ and $\delta_{ij(k,l)}^{(t)}$ for all t. In the M-step, model parameters are estimated using the following equations from (Xing et al., 2010) based on the principle of maximizing the log-likelihood from the E-step:

$$\hat{\beta}_{k,l} = \frac{\sum_{i,j} e_{ij}^{(t)} \delta_{ij(k,l)}^{(t)}}{\sum_{i,j} \delta_{ij(k,l)}^{(t)}},$$

$$\hat{\Phi} = \frac{1}{T-1}\left(\sum_{t=1}^{T-1}(\hat{\mu}_{t+1|T} - \hat{\mu}_{t|T})(\hat{\mu}_{t+1|T} - \hat{\mu}_{t|T})^T + \sum_{t=1}^{T-1} L_t P_{t+1|T} L_t^T\right),$$

$$\hat{\Sigma}^{(t))} = \frac{1}{N}\left(\sum_i (\hat{\mu}_{t|T} - \tilde{\gamma}_i^{(t)})(\hat{\mu}_{t|T} - \tilde{\gamma}_i^{(t)})^T + \sum_i \tilde{\Sigma}_i^{(t)}\right),$$

$$\hat{\nu} = \hat{\mu}_{1|T}$$

Because the variational cluster marginals $q(\vec{z}_{i\to j}, \vec{z}_{j\leftarrow i})$, $q(\vec{\gamma}_i)$, and $q(\vec{\mu}^{(1)}, \ldots, \vec{\mu}^{(T)})$ each depend on variational parameters that are themselves defined by other cluster marginals, the overall algorithm can be seen as a fixed-point iteration that will converge to a local optimum. In order to find a near-global optimum, (Xing et al., 2010) used multiple random restarts to encourage convergence outside a local optimum. We utilized this methodology, as expanded on in Section 3.

We note that the model presented in (Xing et al., 2010) does not capture the relationship between $\vec{\pi}_i^{(t)}$ and $\vec{\pi}_i^{(t+1)}$ directly. The time-evolving nature of $\vec{\mu}$ is modeled, but the corresponding $\vec{\gamma}_i^{(t)}$ is drawn from a distribution parametrized by $\vec{\mu}$, so there is no guarantee that $\vec{\gamma}_i^{(t)}$ will be similar to $\vec{\gamma}_i^{(t-1)}$, and therefore we do not see much linkage between $\vec{\pi}_i^{(t)}$ and $\vec{\pi}_i^{(t+1)}$. Other models, such as the Co-evolving Mixed Membership Blockmodel (CEMMSB) directly parametrize $\vec{\pi}_i$ with $\vec{\mu}$ so that the time-evolving nature of $\vec{\mu}$ is reflected directly in $\vec{\pi}_i$. Therefore one potential improvement for the method is to incorporate the temporal connection not only for the overall distribution of the mixed membership but also more specifically for each node.

2.3. Spectral Clustering

A frequentist method that can be used to estimate dMMSB is spectral clustering. Given an adjacency matrix A for n nodes, we can calculate the eigenvalue decomposition of A such that

$$A\xi_j = \lambda_j \xi_j, \text{ for } j = 1, \ldots, n$$

with

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$$

Then, for k clusters, we compute $k-1$ entry-wise ratios

$$R = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_{k-1}] \in \mathbb{R}^{n\times(k-1)}$$

such that

$$\tilde{r}_j = \frac{\xi_{j+1}}{\xi_j}, \text{ for } j = 1, \ldots, k-1$$

Notice that if we define $r_1, \ldots, r_n$ as the rows of R such that

$$R = [r_1^T, \cdots, r_n^T], r_i \in \mathbb{R}^{k-1}$$

then $r_1, \ldots, r_n$ are just embeddings of the n points in $k-1$ dimensions (see Figure 2). In the noiseless case, these embeddings form a perfect k-simplex in $\mathbb{R}^{k-1}$. If noise is present, the embeddings will form a perturbed k-simplex in $\mathbb{R}^{k-1}$.

From this embedding in $\mathbb{R}^{k-1}$, we use standard vertex hunting methods to find k vertices of the k-simplex, $v_1, \ldots, v_k$. Then for each r_i, there exist weights $w_{i,1}, \ldots, w_{i,k}$ with

$$\sum_{j=1}^k w_{i,j} = 1, \text{ and } r_i = w_{i,1}v_1 + \ldots + w_{i,k}v_k$$

such that $w_i \propto b_1 \cdot \vec{\pi}_i$, where b_1 is related to the size of each cluster and can be calculated from λ and v.

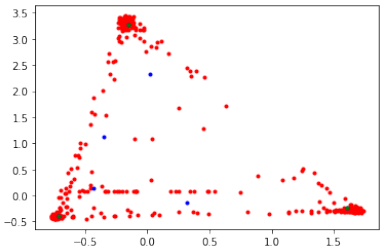


Figure 2. Example of embedding of n nodes in 2 dimensions produced using the frequentist approach. Each red point is an embedding of a node in the network, and each green point is a vertex calculated using vertex hunting. The result is a slightly perturbed 3-simplex due to noise.

Algorithm	Benefits	Drawbacks
Variational EM	Efficiently estimates $\hat{B}$ well	Computationally expensive to estimate $\hat{\gamma}$ well
Spectral Clustering	Efficiently estimates $\hat{\gamma}$ well	Coarse estimate for $\hat{B}$

Table 1. The benefits and drawbacks of the Bayesian Variational EM algorithm and the spectral clustering algorithm seem well poised to balance each other, leading us to believe that ensemble methods can allow for efficient and accurate estimate of $\hat{B}$ and $\hat{\gamma}$

Our experiments reveal that the spectral clustering algorithm is able to estimate the membership of each vertex quite well. However, the algorithm is unable to estimate the role-compatibility matrix B accurately. We attribute this partially to the fact that the vertex hunting algorithm only makes one pass over the data, so there is no re-sampling process to mitigate the effects of noise like there is for the EM algorithm. Nevertheless, the single pass over the data does allow the spectral clustering algorithm to return results much faster than the Bayesian approach.

2.4. Ensemble of EM and Spectral Clustering

There are few existing implementations that incorporate both the frequentist spectral clustering approach and the Bayesian EM approach. We see great promise in ensemble methods because the benefits and drawbacks of each approach seem to balance each other out.

For example, the EM algorithm requires many epochs to converge to the global minimum in the loss surface for accurate estimation of the mixed-membership vectors $\vec{\pi}_i$. It also requires multiple random restarts because the algorithm is prone to getting stuck in local optima. The number of iterations required to estimate $\vec{\pi}_i$ well can be vastly reduced by initializing the algorithm with values that are closer to the global minimum in the loss surface. In order to accomplish this, we run the spectral clustering algorithm to get good estimates of $\vec{\pi}_i$ and a poor estimate of B. We can then use these estimates to initialize the EM algorithm for the LNMMSB model and compute a much better estimate of B (which takes many fewer iterations than calculating estimates of $\vec{\pi}_i$ with random initialization).

Another option is to run the EM algorithm for a small number of iterations to get a good estimate of B only and then run the spectral clustering algorithm to get a good estimate of γ. While both of these estimates should be reasonably good, the cluster labels returned from each algorithm will not match. Thus, an extra step of permuting the cluster labels to determine which of the $k!$ labellings results in the best log-likelihood is required to combine the results from both algorithms and match the labellings at different timesteps.

3. Experiments and Conclusions

3.1. Study of LNMMSB

Even without the temporal structure, it is not easy to estimate the parameters in an LNMMSB accurately. We found this by trying two implementations on Github [1][2] (there are few python implementations). They are for general MMSB, but do not work well on our simulated data even if we generate the networks from MMSB. For our own implementation of the variational EM algorithm in (Xing et al., 2010), there were also many numerical issues at the beginning, and we spent much time to overcome those issues.

To understand the Log-Normal Mixed Membership Block model better, we investigate the estimation error of the connectivity matrix B, the membership matrix Π, and the labels $\{C_i\}$ under different prior on γ, using a publicly available spectral clustering method called Mix-SCORE (Jin et al., 2017). We set $\gamma \sim \text{Normal}(0, \sigma^2 I_K)$, and let σ^2 vary from 1 to 10000. The results in Figure 3 suggest that the estimation of B, Π, and $\{C_i\}$ can be very hard for small values of σ^2.

The intuition behind this phenomenon is shown in the top panel of Figure 3: as σ^2 gets smaller, the membership hyperparameters γ_i concentrate more around $\mu = 0$, and consequently, the membership vectors π_i will concentrate more around $(0.5, 0.5)$, meaning that the two clusters get mixed more and harder to estimate. For general μ, the issue is similar, because if π_i concentrates around a point close to $(0, 1)$ or $(1, 0)$, the majority of nodes will be in one cluster, making the overall estimation harder. "harder" here means that for spectral clustering, it will give worse estimates of B and π, while for variational EM, it will take more time to give good enough estimates.

To test the validity of our implemented variational EM algorithm, we tried to reproduce the results in Figure 2 in

[1]https://github.com/aburnap/Mixed-Membership-Stochastic-Blockmodel

[2]https://github.com/dongwookim-ml/MMSB

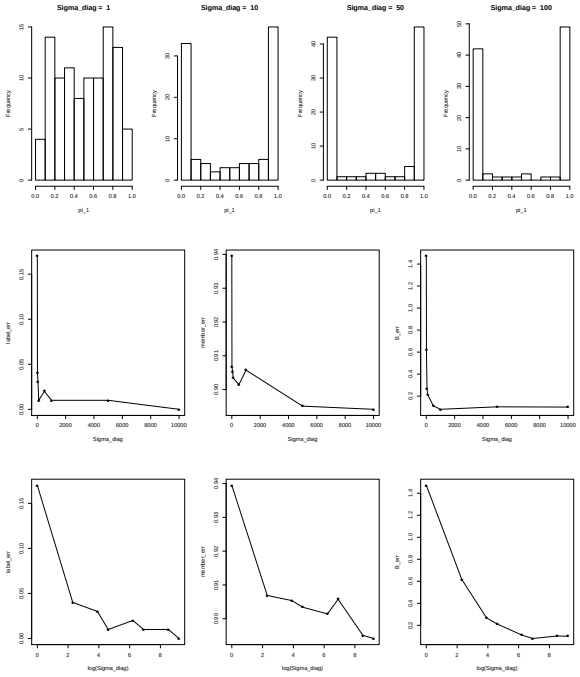


Figure 3. Top: the distribution of membership $\pi_i(1)$. Mid, Bottom: Estimation error of the the labels $\{C_i\}$, the membership matrix Π, and the connectivity matrix B.

(Xing et al., 2010). It turns out that our model can estimate the 3 B matrices very well. Figure 4 shows the path of the log-likelihood $\ell(E|B)$, the approximated lower bound $\hat{\ell}(E|B)$ of the log-likelihood used in the variational EM algorithm, and the l_1 distance between $\theta^{(s-1)}$ and $\theta^{(s)}$ when the inner loop stops. In the experiment, we initialize B and μ uniformly from $[0, 1]$.

Table 2 compares the estimates $\hat{B}$ and the truth B, showing that our implementation is doing as good as the one for that paper.

true			est			true			est		
1.0	0.3	0	1.0	0.30	0	0.45	0	0.05	0.51	0	0
0.3	1.0	0	0.29	1.0	0	0	0.50	0	0	0.50	0
0	0	1.0	0	0	0	0	0	0.40	0	0	0.42

Table 2. Ground truth and estimates of the connectivity matrix B for a static network (LNMMSB) calculated using the variational EM algorithm. The result is as good as the one in Fig 2 of (Xing et al., 2010)

Then we compare the performance of the three methods: EM, spectral clustering, and the ensemble of EM and spectral clustering in Figure 5 and Table 3. It should be noted that there are three several parameters when fitting the EM algorithm: maximum outer iterations, maximum inner iterations, the number of re-initializations and the tolerances of convergence for outer and inner loops, so we indicate the extent of fitting simply by the time it takes to run the algorithm. It can be seen in Figure 5 that when the the EM algorithm is run for too short time, the estimation of π can be very coarse (panel (a)), while Table 3 shows that the $\hat{B}$ is fairly accurate (left).

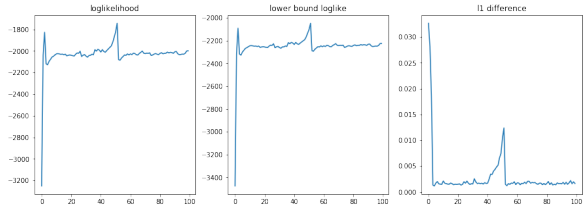
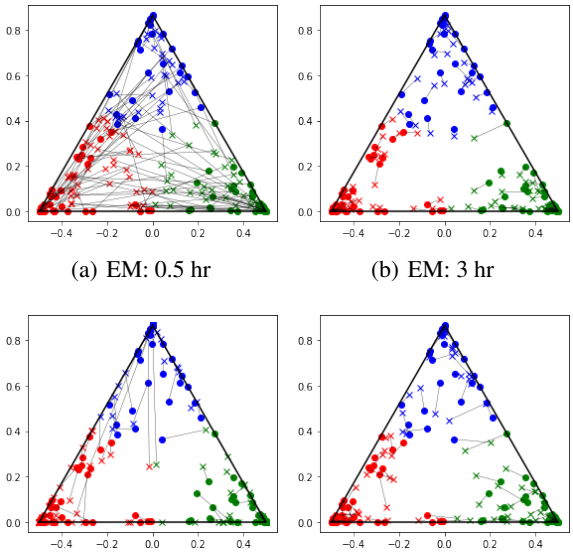


Figure 4. The training curves of the EM algorithm for estimating LNMMSB corresponding to panel (b) in Figure 5



(a) EM: 0.5 hr (b) EM: 3 hr

(c) Spectral: $<$ 1min (d) EM+Spectral: 6 min

Figure 5. A comparison of π and $\hat{\pi}$ calculated using the variational EM approach, the spectral clustering approach, and the ensemble approach consisting of both EM and spectral clustering.

3.2. Study of dMMSB

We compare the performance of the three methods on estimating $\hat{B}$ and $\hat{\pi}$ in Table 4, in a simple case where $N = 100$, $K = 3$, $T = 4$, $\nu = 0$, $\Phi = 0.01I_3$, and $\Sigma = 10I_3$. The connectivity matrix $B = I_3 + 0.3E_{12} + 0.3E_{21}$ where E_{ij} is a matrix whose all entries are 0 except $E_{ij}(i, j) = 1$. The results again support our assertion that the ensemble of variational EM and spectral clustering can give accurate estimates of B and π with much less computational price than variational EM itself.

EM (0.5 hr)		
1.0	0.24	0.0
0.27	1.0	0.0
0.0	0.0	1.0

Spectral (< 1 min)		
0.7188	0.3407	0.1768
0.3407	0.7173	0.1968
0.1768	0.1968	0.6688

EM + Spectral (6 min)		
0.9963	0.2465	0.0
0.2465	0.9963	0.0
0.0	0.0	0.9438

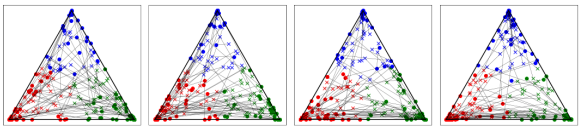
Table 3. Ground truth and estimates of the connectivity matrix B for a static network (LNMMSB) calculated using spectral clustering the ensemble approach consisting of both variational EM and spectral clustering. These results were achieved in only 5 minutes. The spectral clustering algorithm is not able to estimate B well, but when its results are used to initialize the variational EM algorithm the estimate of B vastly improves and can be computed with only a few iterations.

EM (12 hr)		
1.0	0.30	0
0.31	1.0	0
0	0	1.0

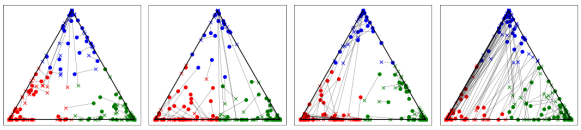
Spectral (< 1 min)		
0.67	0.26	0.21
0.25	0.66	0.32
0.22	0.32	0.65

EM+Spectral (12 min)		
1.0	0.21	0
0.22	1.0	0.02
0	0.01	1.0

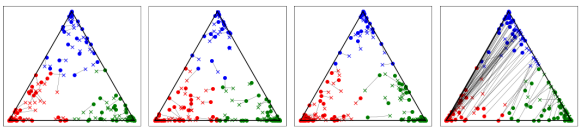
Table 4. Comparison of the estimator $\hat{B}$ of the three methods on simulated data from dMMSB where $B = I_3 + 0.3E_{12} + 0.3E_{21}$.



(a) EM (12 hr)



(b) Spectral (< 1 min)



(c) EM + Spectral (12 min)

Figure 6. Comparison of the estimator $\hat{\pi}$ of the three methods on simulated data from dMMSB. t = 0,1,2,3 from left to right.

3.3. Applying the method to gene coexpression networks with temporal and spatial diversity

Gene coexpression networks can be utilized as a method to detect the genes within regulatory pathways or gene communities for a cell type. To understand cell development or maturation, it is helpful for us to study the dynamic changing of gene coexpression spatially and temporarily, which suggests how genes interact and function in different cell types at different stages. By clustering the network, we are able to identify which genes might belong to a same gene community which might cofunction.RNA-Seq is a widely used biological technique to measure the expression level in cells accurately by sequencing the mRNA transcripts another RNA segments.

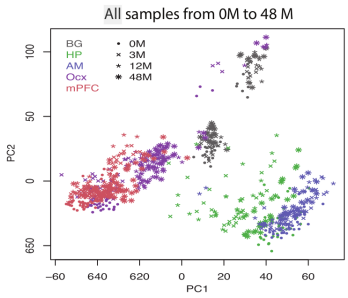


Figure 7. First two principal components for samples from 0 to 48 Months, from (Liu et al., 2018)

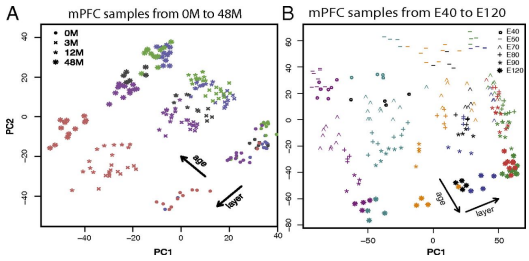


Figure 8. First two principal components for samples in mPFC region from all ages, from (Liu et al., 2018)

However, most of the research cluster the gene expression at different timepoint separately, which might lose certain information regarding the same cell type. (Liu et al., 2018) used global spectral clustering to detect temporal changes in gene coexpression network from transcriptomes in samples of rhesus monkey brains of different ages and layer conducted in (Bakken et al., 2016). This dataset contains expression levels of 12441 genes from 1936 samples obtained from different ages including six prenatal periods (40 to 120 Embryonic days) and four postnatal period (0 to 48 Months) and different regions and layers of rhesus monkey brain. From the top two principal components plot for both samples with age 0 to 48 months and samples in mPFC region in different layers from all ages, the age and region of the brain are the main two factors which explain the variances.

We used samples in mPFC regions in postnatal phases to test the temporal (the embryonic ages) change of community composition.

Construct networks For simplicity, we only consider the temporal index when constructing the series of networks. For each of the 10 time periods, we calculate the correlations between the expression level of genes, and a correlation that is at least 0.7 is determined as an edge in the network.

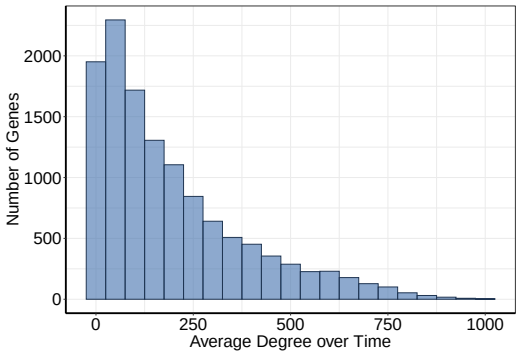
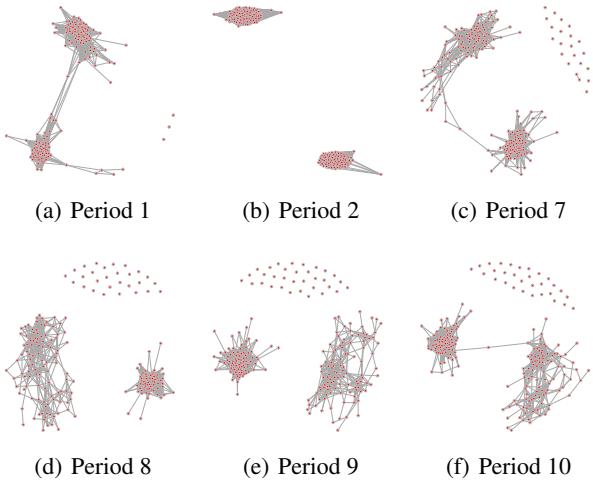


Figure 9. Empirical distribution of the average degrees of genes over 10 time periods.

Cut off genes The number of genes is huge for a network to perform clustering algorithms efficiently. The constructed networks are very sparse, as the degrees of most genes in the network are very small compared to the huge size of the network, shown in Figure 9. Therefore we select only those genes with average degrees at least 700. This operation reduces the size of the network from 12441 to 266.

Exploratory analysis We visualize the networks in the 10 periods in Figure 10. It is naturally clustered into two big subgoups for each period. Thus for convenience we chose $k = 2$ for the membership number.



(a) Period 1 (b) Period 2 (c) Period 7

(d) Period 8 (e) Period 9 (f) Period 10

Figure 10. Empirical distribution of the average degrees of genes over 10 time periods. There is an edge between two genes if the correlation of them is at least 0.7. We omit the networks from period 3 to period 6 as they all look like the network in period 2.

Apply both methods to networks We apply both variational EM for dMMSB and spectral clustering to the expression network data for samples with four selected age groups (0M, 3M, 12M, 48M). The results show that for dynamic network with changes of membership, the variational EM of dMMSB outperforms the spectral clustering. As we can see, in 3M and 12M, in the result from spectral clustering, the cluster red arrow pointed are predicted as almost a mixture of both membership. However, from dMMSB, the cluster blue arrow pointed are predicted a same membership, which is consistent with the visualization. This suggest that variational EM method which takes the temporal information into account can improve the temporal network clustering accuracy in real case scenario.

We yet have time to dig into what specific genes change the membership temporally. But it certainly can give us some insight on the gene cofunction and upstream downstream pathways during postnatal development.

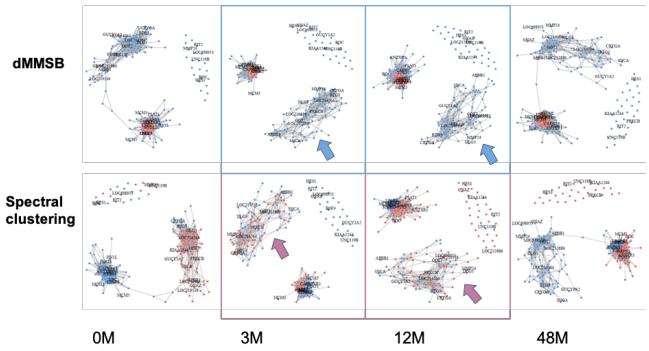


Figure 11. Result comparison of both variational EM and spectral clustering for dMMSB with ages 0M, 4M, 12M and 48M. Top 200 genes with highest degrees are chosen. $k = 2$

4. Conclusion

Clustering network is a hard problem. Introducing temporal structure makes the problem even harder. MMSB model allows a node in the network to have mixed membership. dMMSB is more powerful for its inclusion of temporal information of the network. Variational EM based on dMMSB (Xing et al., 2010) can cluster dynamic networks in a temporal view, which can provide an accurate estimation on certain parameters but with a high computational expense. Spectral clustering on the other hand is a more efficient method to cluster network without the model assumption. It can estimate a quite accurate γ but a coarse B. There is no public available implementation for variational EM based on dMMSB and the spectral clustering algorithm we used. Therefore we implemented the methods on our own. Our results on simulation data clearly show the advantages and disadvantages of either method. For gene expression network, we observe an outperformance of variational EM for dMMSB on spectral clustering. We also managed to ensemble both methods so that the estima-

tion can be more efficient and not lose accuracy.

References

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. Mixed membership stochastic blockmodels. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 21*, pp. 33–40. Curran Associates, Inc., 2009. URL https://dl.acm.org/doi/pdf/10.5555/1390681.1442798.

Bakken, T. E., Miller, J. A., Ding, S. L., Sunkin, S. M., Smith, K. A., Ng, L., Szafer, A., Dalley, R. A., Royall, J. J., Lemon, T., Shapouri, S., Aiona, K., Arnold, J., Bennett, J. L., Bertagnolli, D., Bickley, K., Boe, A., Brouner, K., Butler, S., Byrnes, E., Caldejon, S., Carey, A., Cate, S., Chapin, M., Chen, J., Dee, N., Desta, T., Dolbeare, T. A., Dotson, N., Ebbert, A., Fulfs, E., Gee, G., Gilbert, T. L., Goldy, J., Gourley, L., Gregor, B., Gu, G., Hall, J., Haradon, Z., Haynor, D. R., Hejazinia, N., Hoerder-Suabedissen, A., Howard, R., Jochim, J., Kinnunen, M., Kriedberg, A., Kuan, C. L., Lau, C., Lee, C. K., Lee, F., Luong, L., Mastan, N., May, R., Melchor, J., Mosqueda, N., Mott, E., Ngo, K., Nyhus, J., Oldre, A., Olson, E., Parente, J., Parker, P. D., Parry, S., Pendergraft, J., Potekhina, L., Reding, M., Riley, Z. L., Roberts, T., Rogers, B., Roll, K., Rosen, D., Sandman, D., Sarreal, M., Shapovalova, N., Shi, S., Sjoquist, N., Sodt, A. J., Townsend, R., Velasquez, L., Wagley, U., Wakeman, W. B., White, C., Bennett, C., Wu, J., Young, R., Youngstrom, B. L., Wohnoutka, P., Gibbs, R. A., Rogers, J., Hohmann, J. G., Hawrylycz, M. J., Hevner, R. F., Molnár, Z., Phillips, J. W., Dang, C., Jones, A. R., Amaral, D. G., Bernard, A., and Lein, E. S. A comprehensive transcriptional map of primate brain development. *Nature*, 2016. ISSN 14764687. doi: 10.1038/nature18637.

Cho, Y.-S., Steeg, G. V., and Galstyan, A. Mixed membership blockmodels for dynamic networks with feedback. In *Handbook of Mixed Membership Models and Their Applications*, 2014.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002. doi: 10.1198/016214502388618906. URL https://doi.org/10.1198/016214502388618906.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983. ISSN 0378-8733. doi: https://doi.org/10.1016/0378-8733(83)90021-7. URL http://www.sciencedirect.com/science/article/pii/0378873383900217.

Jin, J., Ke, T. Z., and Luo, S. Estimating network memberships by simplex vertex hunting. *arXiv:1708.07852*, 2017.

Kim, B., Lee, K. H., Xue, L., and Niu, X. A review of dynamic network models with latent variables. *Statistics surveys*, 12:105, 2018.

Kolar, M., Song, L., Ahmed, A., and Xing, E. P. Estimating time-varying networks. *Ann. Appl. Stat.*, 4(1):94–123, 03 2010. doi: 10.1214/09-AOAS308. URL https://doi.org/10.1214/09-AOAS308.

Liu, F., Choi, D., Xie, L., and Roeder, K. Global spectral clustering in dynamic networks. *Proceedings of the National Academy of Sciences*, 115(5):927–932, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1718449115. URL https://www.pnas.org/content/115/5/927.

Pensky, M. and Zhang, T. Spectral clustering in the dynamic stochastic block model. *Electron. J. Statist.*, 13 (1):678–709, 2019. doi: 10.1214/19-EJS1533. URL https://doi.org/10.1214/19-EJS1533.

Perry, P. O. and Wolfe, P. J. Point process modelling for directed interaction networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(5):821–849, 2013. doi: 10.1111/rssb.12013. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/rssb.12013.

Rastelli, R., Friel, N., and Raftery, A. E. Properties of latent variable network models. *Network Science*, 4(4):407–432, 2016.

Sarkar, P. and Moore, A. W. Dynamic social network analysis using latent space models. In Weiss, Y., Schölkopf, B., and Platt, J. C. (eds.), *Advances in Neural Information Processing Systems 18*, pp. 1145–1152. MIT Press, 2006. URL https://doi.org/10.1145/1117454.1117459.

Xing, E. P., Fu, W., and Song, L. A state-space mixed membership blockmodel for dynamic network tomography. *Ann. Appl. Stat.*, 4(2):535–566, 06 2010. doi: 10.1214/09-AOAS311. URL https://doi.org/10.1214/09-AOAS311.

Xu, K. and Hero, A. Dynamic stochastic blockmodels for time-evolving social networks. *Selected Topics in Signal Processing, IEEE Journal of*, 8:552–562, 03 2014. doi: 10.1109/JSTSP.2014.2310294.