Inferring Block Structure of Graphical Models in Exponential Families

Siqi Sun* Hai Wang* Jinbo Xu
Toyota Technological Institute at Chicago, USA 60637
{siqi.sun, haiwang, j3xu}@ttic.edu

Abstract

Learning the structure of a graphical model is a fundamental problem and it is used extensively to infer the relationship between random variables. In many real world applications, we usually have some prior knowledge about the underlying graph structure, such as degree distribution and block structure. In this paper, we propose a novel generative model for describing the block structure in general exponential families, and optimize it by an Expectation-Maximization (EM) algorithm with variational Bayes. Experimental results show that our method performs well on both synthetic and real data. Furthermore, our method can predict overlapping block structure of a graphical model in general exponential families.

1 INTRODUCTION

Graphical models are an important tool to describe and model real world data in various fields such as nature language processing, computation biology and image analysis. Learning the structure of a graphical model is thus essential since it provides a convenient way to model conditional independence among variables for further analysis. Plenty of work has been done for some specific settings of this structure learning problem. For Gaussian Graphical Models (GGMs), it is well known that conditional independence is encoded in the precision matrix. Neighborhood estimation [19] and log-likelihood maximization with $l_1$ penalty [5, 11, 30] have been developed to estimate the structure of GGMs. Furthermore, for the Ising, Poisson and other models in exponential families, a consistent neighborhood estimator is proposed by Ravikumar et al [24] and Yang et al [29], which apply a logistic regression and a generalized linear model with $l_1$ penalty, respectively, to learn the underlying graph structure.

Recently, incorporating prior knowledge into structure learning has drawn much attention because the graph under estimation in many real-world applications usually has some intrinsic properties, such as scale free [9, 16], block structure [3, 17, 22, 26] and other topological constraints [10]. Amongst those properties, the block structure of a graph is of special interest for biological networks. For example, in a protein-protein interaction network, proteins are more likely to form a pathway or complex to express a specific function [25]. It is straightforward to estimate the graph first, and then apply a clustering algorithm such as spectral clustering [27, 28] or Mixed-Membership Stochastic Blockmodel (MMSB) [1, 12, 13] to obtain block structure. However, simultaneously inferring the graph and block structure may improve the result in terms of both cluster accuracy and graph estimation [22, 26], especially when the data is limited. To the best of our knowledge, existing works that focusing on block structure in graphical models are all based on Gaussian graphical models [3, 15, 17, 18, 26] or factor models [22], and can only be used to infer hard clustering and non-overlapping block structure.

This paper proposes a generative model that can (1) apply to a graphical model on exponential families, (2) infer the underlying soft clusters as well as overlapping blocks, and (3) generate graph and block structure at the same time with better accuracy than those generated by a heuristic approach. The reason why we want to infer them together is that the hidden block structure can actually help decide the penalty parameters. For example, we might increase the penalty parameters if two nodes are from different blocks because the probability that there is an edge between them is small, and also for the converse. In the rest of the paper, we first describe some related work and notations in Section 2, and then present our method in Section 3. Finally, the results for synthetic data and a microRNA network data and the conclusions are presented in Section 4 and 5, respectively.

*These two authors contributed equally to this work.
2 RELATED WORK & NOTATION

Let \( x = (x_1, x_2, \ldots, x_p) \) be a random vector from a multivariate Gaussian distribution, \( X_{n \times p} \) the observed data, and \( n \) the number of samples. It has been proved that \( x_i \) and \( x_j \) are conditionally independent if and only if \( \Omega_{ij} = 0 \), where \( \Omega \) is the precision matrix of the Gaussian distribution \([19]\).

We can also use a graph \( G = (V, E) \) to represent the relationship among the \( p \) random variables \( x_1, x_2, \ldots, x_p \) where \( V \) and \( E \) are the vertex and edge sets, respectively. Meanwhile, \( V \) is the set of the \( p \) variables and \( E \) models conditional dependency between the variables. To obtain a sparse graph, some methods have been proposed to estimate the sparse precision matrix \([5, 11, 19, 30]\) by maximizing the \( l_1 \) penalized log-likelihood, i.e.,

\[
\hat{\Omega} = \arg \max_{\Omega > 0} \log \det(\Omega) - tr(\Omega \Sigma) - \lambda \sum_{i \leq j} |\Omega_{ij}|, \tag{1}
\]

where \( \lambda \) is the regularizer that controls the sparsity level and \( \Sigma \) is the empirical covariance matrix. The formulation defined in Eq. (1) can be efficiently optimized by graphical lasso \([11]\).

Another line of work first estimates the neighborhood \( \hat{N}(x_a) \) for each random variable \( x_a \), i.e. node \( a \), and then constructs the graph as the union of all neighborhoods.

In GGMs, the neighborhood of one node can be estimated by Lasso as follows,

\[
\hat{w}_a = \arg \min_{w} \frac{1}{n} \|X_a - X_a w_a\|_2^2 + \lambda \|w_a\|_1, \tag{2}
\]

where \( X_a \) is the \( a \)-th column of \( X \) and \( X_{-a} \) is the whole data matrix excluding column \( a \).

More generally, consider the graphical model where its conditional distribution can be written as an exponential form with linear interactions, i.e.,

\[
P(x_a|x_{-a}, w_a) = q_0(x_a) \exp(\sum_b w_{ab} x_a x_b - D(x_{-a}, w_a)), \tag{3}
\]

where \( x_{-a} \) indicates all random variables except \( x_a \), \( q_0(x) \) is the base measure, \( w_a \) is the model parameter and \( D(x_{-a}, w_a) \) is the log-normalizing constant. Given the observed data \( X_{n \times p} \), the conditional log-likelihood is

\[
l(X_a, w_a) = \frac{1}{n} \sum_{i=1}^{n} \sum_{b \neq a} w_{ab} x_i^a x_i^b + D(w_a, X_{-a}), \tag{4}
\]

and its \( l_1 \) regularized estimator is

\[
\hat{w}_a = \arg \min_{w_a} l(X_a, w_a) + \lambda_a \|w_a\|_1. \tag{5}
\]

The neighbor of \( a \) can then be estimated as \( \hat{N}(a) = \{b \in V/a : \hat{w}_{ab} \neq 0\} \). The problem (5) is usually a linear model (in GGMs) or a generalized linear model (in Ising or Potts model) with \( l_1 \) penalty, which can be solved efficiently by methods like iterative soft-thresholding \([7]\). The advantage of this estimator is that it has both sparsity and consistency \([19, 24, 29]\). Further it can be easily implemented and parallelized, so it is scalable for very large scale data.

For graphical models with block structure, most of existing work focused on GGMs. Marlin et al \([17]\) proposed a two-stage Bayesian model, in which a spike and slab like prior based on network with block structure is applied to generate model parameters \((w)\), and the data is then generated by linear regression given \( w \). The posterior is optimized by variational inference.

Ambroise et al \([3]\) used hidden indicator variables \( Z \) to denote the cluster assignment, where only one element of \( Z \) is equal to one, and all the others are zero. Then the precision matrix is estimated by maximizing the log complete-data likelihood spreading over \( Z \). Since summing up all configurations of \( Z \) is intractable, an EM algorithm is employed to perform the inference.

Determining the number of clusters \( K \) is still challenging, and the two previous methods use a heuristic split approach and ICL (integrated complete likelihood) criterion respectively. To solve this issue, Marlin et al \([18]\) introduced a novel prior and performed MAP estimation on the model so that the model can automatically determine the number of blocks. Furthermore, Sun et al \([26]\) proposed a Bayesian method that used Chinese Restaurant Process and Wishart prior to model the number of clusters and precision matrix \( \Omega \), respectively, and Gibbs sampling to estimate the posterior of block membership variables. Besides, Palla et al \([22]\) proposed a nonparametric Bayesian method to cluster variables in factor models. Note that nearly all of these work only focus on GGMs, and assume a non-overlapping or hard clustering block structure.

3 Methods

In this section we present a generative process to model the dependency between random variables. Assuming that the number of clusters \( K \) is known, our method can be briefly described as follows. We first use MMSB to generate a block structured network. That is, we sample the hidden variables \( \Theta \) from some prior distribution \( P(\Theta|\alpha, \eta) \) specified by MMSB, where \( \alpha \) and \( \eta \) are hyper parameters. Then given \( \Theta \), we sample the model parameters \( w \) from a Laplace prior \( P(w|\Theta) \). Finally, we sample the data from \( P(X|w) \) in a pseudo-likelihood manner.
3.1 A Generative Model

We describe our model in a top-down fashion:

- Sample cluster strength $\beta_k \sim \text{Beta}(\eta)$ for each cluster $1, \ldots, K$.
- Sample cluster membership $\pi_a \sim \text{Dirichlet}(\alpha)$ for each node $a = 1, \ldots, p$.
- For each pair of nodes $a$ and $b$,
  - Sample interaction indicator $z_{a \rightarrow b} \sim \pi_a$.
  - Sample interaction indicator $z_{b \rightarrow a} \sim \pi_b$.
  - Compute $r_{ab} = \beta_1^{I(z_{a \rightarrow b} = 1)} \epsilon^{1-I(z_{a \rightarrow b} = 0)}$.
  - Sample $w_{ab} \sim \text{Laplace}(\rho_{ab}(r_{ab}))$.
- For each node $a$,
  - Fit the data by generalized linear model, i.e. $x_a | x_{-a} \sim q_0(x_a) \exp(\sum_{b \neq a} w_{ab} x_a x_b - D(x_{-a}, w_a))$.

where $\epsilon$ is the probability that there is an edge between different clusters. $z_{a \rightarrow b}, z_{b \rightarrow a}$ are $K$ dimension indicator vectors, i.e. $z^k_{a \rightarrow b} = 1$ means node $a$ is in cluster $k$.

Based upon the proposed model above, we can write down the complete data likelihood $P(X, w, \Theta)$.

**Proposition 1.** The complete data likelihood of the model described above can be written as

$$P(X, w, \Theta) = P(X|w)P(w|\Theta)P(\Theta)$$

$$= \prod_{i=1}^{n} \exp\left(\sum_{a=1}^{p} \sum_{b=1}^{p} w_{ab} X_a^i X_b^i + C(X_a) - D(X_{-a})\right)$$

$$\prod_{a,b} \frac{1}{2\rho_{ab}(r_{ab})} \exp\left(-\frac{|w_{ab}|}{\rho_{ab}(r_{ab})}\right)$$

$$\prod_{k} \frac{1}{B(\eta_k, \eta_k)} \beta_k^{\eta_k-1}(1 - \beta_k)^{\eta_k-1}$$

$$\prod_{a \leq b} \prod_{k} \pi_{a \rightarrow b}^{\pi_k} \pi_{b \rightarrow a}^{\pi_k} \prod_{a} \frac{1}{B(\alpha_k)} \prod_{k} \pi_{a \rightarrow k}^{\alpha_k-1}. \quad (6)$$

where $B(\eta, \eta)$ is a Beta function and $B(\alpha) = \prod_{i=1}^{K} \frac{\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$ is the multivariate Beta function. The proof is obvious according to Bayes rule.

3.2 Optimization

We estimate $w$ by maximizing its posterior given data $X$, i.e.

$$\hat{w} = \arg \max_{w} \log P(w|X) = \arg \max_{w} \log P(X, w). \quad (7)$$

However, the distribution of $w$ depends on the hidden variable $\Theta$, which encodes all the cluster structure information. If we marginalize over the hidden variables $\Theta$, we obtain

$$\hat{w} = \arg \max_{w} \sum_{\Theta} P(X, w, \Theta), \quad (8)$$

where the summation is intractable because it is over all possible values of the hidden variables. Here we introduce an EM algorithm to optimize it. To do so we need to compute the following conditional expectation:

$$Q(w|w^f) = \mathbb{E}_{\Theta|X, w^f}[\log P(X, w, \Theta)]$$

$$= \sum_{\Theta} P(\Theta|w^f) \log P(X, w, \Theta). \quad (9)$$

Unfortunately, the calculation of $Q(w|w^f)$ is still intractable since $P(\Theta|w^f)$ cannot be factorized. To deal with this, we first use variational approach to approximate $P(\Theta|w^f)$ and then the classical EM approach.

3.2.1 Expectation Step

In the E step, the parameter $w$ is assumed to be known as $w^f$. Here we need to approximate the posterior of $\Theta$ by its
Supposing that \( Q(w|w^t) \) can be approximated by
\[
Q(w|w^t) = \sum_{\Theta} q(\Theta) \log P(X, w, \Theta).
\]

To simplify the calculation, we approximate the Laplace distribution \( P(w_{ab}(\Theta)) \) by discretizing \( w_{ab} \) into a binary variables \( \hat{w}_{ab} \). More specifically, we approximate \( P(\Theta|w^t) \) as
\[
P(\beta, \pi, z|w) \approx \frac{1}{P(w)} P(w|z, \beta) P(z|\pi) P(\pi|\alpha) P(\beta|\eta)
\]
\[
\approx \prod_k q(\beta_k | \lambda_k \pi) \prod_a q(\pi_a | \gamma_a)
\]
\[
\prod_a q(z_{a\rightarrow b} | \phi_{a\rightarrow b}) q(z_{a\leftarrow b} | \phi_{a\leftarrow b}), \quad (10)
\]
where \( q(\cdot) \) is variational distribution, and \( \lambda, \gamma, \phi \) are variational parameters. To narrow the gap between its variational approximation and posterior distribution, we maximize the evidence lower bound (ELBO), which is equivalent to minimizing the KL divergence. To prove the equivalence, note that
\[
KL(q(\Theta)||P(\Theta|w^t)) = \sum_{\Theta} q(\Theta) \log \frac{q(\Theta)}{P(\Theta, w_t)} + \log P(w_t),
\]
where the first term in the right hand side of Eq. (11) is negative ELBO.

**Proposition 2.** Supposing that \( q(\Theta) \) can be decomposed according to Eq. 10, the approximated ELBO (denoted as \( L \)) can then be written as:
\[
L \approx \sum_k E_q[\log p(\hat{w}|z, \beta)] P(z|\pi) P(\pi|\alpha) P(\beta|\eta)
\]
\[
- \sum_k E_q[\log q(\beta_k | \lambda_k \pi) q(z|\phi) q(\pi|\gamma)]
\]
\[
+ \sum_a E_q[\log p(\pi_a | \alpha) - \sum_a E_q[\log q(\pi_a | \gamma_a)]
\]
\[
+ \sum_{a,b} E_q[\log p(z_{a\rightarrow b} | \pi_a) + E_q[p(z_{a\leftarrow b} | \pi_b)]
\]
\[
- \sum_{a,b} E_q[\log q(z_{a\rightarrow b} | \phi_{a\rightarrow b})] + E_q[\log q(z_{a\leftarrow b} | \phi_{a\leftarrow b})]
\]
\[
+ \sum_{a,b} E_q[\log p(\hat{w}_{ab} | z_{a\rightarrow b}, z_{a\leftarrow b}, \beta)]. \quad (12)
\]

For detailed derivation, which will be used for next proposition, please refer to appendix A.

The maximization procedure is usually optimized by coordinate ascent algorithm. However, it would be extremely slow when the number of nodes is large. The reason is that before updating global variables \( \lambda \) and \( \gamma \), we need to compute all \( n^2 \) pairs of local variables \( (\phi_{a\rightarrow b}, \phi_{a\leftarrow b}) \), which is a waste in the first several iterations because the parameters are initialized randomly. Hence we apply stochastic variational inference (SVI) [12, 14] to further speed up computation. In each step of SVI, we use a noisy estimate of gradient from a subsample of nodes for global variables.

**Proposition 3.** Given a pair of nodes \((a, b)\), the estimated gradient for each global variable is
\[
\partial \gamma_{a,k} = \alpha_k + \frac{N(N-1)}{2} \phi_{a\rightarrow b}^k - \gamma_{a,k}
\]
\[
\partial \lambda_{k,i} = \eta_{k,i} + \frac{N(N-1)}{2} \phi_{a\rightarrow b}^k \cdot \phi_{a\leftarrow b}^k \cdot \hat{w}_{ab,i} - \lambda_{k,i},
\]
and the optimal for local variables are
\[
\phi_{a\rightarrow b}^k | \hat{w}_{ab} = 1 \propto \exp \left( \psi(\gamma_{ab}) - \psi(\gamma_{a}) + \phi_{a\rightarrow b}^k \psi(\lambda_{k1}) - \psi(\lambda_{k}) + (1 - \phi_{a\rightarrow b}^k) \log(1 - \epsilon) \right),
\]
\[
\phi_{a\leftarrow b}^k | \hat{w}_{ab} = 0 \propto \exp \left( \psi(\gamma_{ab}) - \psi(\gamma_{a}) + \psi_{a\rightarrow b}^k \psi(\lambda_{k2}) - \psi(\lambda_{k}) + (1 - \phi_{a\leftarrow b}^k) \log(1 - \epsilon) \right),
\]
where \( \psi(x) \) is the digamma function, \( \hat{w}_{ab,1} = \hat{w}_{ab}, \hat{w}_{ab,2} = 1 - \hat{w}_{ab}, \gamma_a = \sum_k \gamma_{ak} \) and \( \lambda_k = \lambda_{k1} + \lambda_{k2} \). The derivation for \( \phi_{a\rightarrow b} \) is similar.

**Sketched Proof:** Following proposition 2 and detailed computation in Appendix, the gradient for \( \gamma_{a,j} \) is
\[
\frac{\partial L}{\partial \gamma_{a,j}} = \sum_{k=1}^K \frac{\partial E_q[\log \pi_{ak}]}{\partial \gamma_{a,j}} \left( \frac{N(N-1)}{2} \phi_{a\rightarrow b,k} + \alpha_k - \gamma_{a,k} \right)
\]
\[
(15)
\]

**Given exponential family**
\[
P(x|w) = q_0(x) \exp(w \cdot T(x) - D(w)),
\]
we have
\[
E_q[\log \pi_{ak}] = \frac{\partial D(\gamma)}{\partial \gamma_{a,k}}
\]
\[
\frac{\partial^2 D(w)}{\partial w_i \partial w_j} = - \frac{\partial^2 D(w)}{\partial w_i \partial w_j}
\]
\[
(17)
\]
Substituting $\frac{\partial E_{q}[\log \pi_{a,k}]}{\partial \gamma_{a,j}}$ in Eq. 15, we have

$$\frac{\partial L}{\gamma_{a,j}} = \sum_{k=1}^{K} \frac{\partial^{2} \log q(\pi_{a}|\gamma_{a})}{\partial \gamma_{a,j} \partial \gamma_{a,k}} \left( \frac{(N-1)}{2} \phi_{a-b,k} + \alpha_{k} - \gamma_{a,k} \right)$$

(18)

After multiplying Eq.18 by the inverse of the Fisher information matrix of $q$, we obtain the update for global variable $\gamma$. The proof for $\lambda$ is similar.

The derivation for local variables is obvious by using the fact that $E[\log \pi_{ak}] = \psi(\gamma_{ak}) - \psi(\gamma_{a})$ if $\pi_{a}$ follows the Dirichlet distribution.

We summarize the E step of our algorithm in algorithm 1. We also use the technique proposed by Gopalan et al [13] to determine the number of blocks (i.e., $K$).

Algorithm 1 Stochastic Variational Inference (E-Step)

Initialize $\gamma_{a}$, $\lambda_{k}$ randomly, $\tau$ is a parameter

for $t = 1$ to MAX ITERATION do

Sample a pair of nodes $(a, b)$ randomly

Compute gradient of local variables based on Eq. 14

Compute gradient of global variables based on Eq. 13

Compute step size $s_{t} = 1/t^{\tau}$

Update global variables by $g \rightarrow g + s_{t} \partial g$

end for

3.2.2 Maximization Step

Now we can approximate $Q(w|w')$ by substituting $P(\theta|w')$ with its variational approximation $q(\theta)$, and infer $w$ from $q(\theta)$. Plugging in the complete loglikelihood in Eq. 6.

$$Q(w|w') \approx \tilde{Q}(w|w') = E_{q(\theta)}[\log P(X|w)P(w|\Theta)P(\Theta)]$$

$$= \sum_{i} \sum_{a,b} w_{ab} X_{a}^{i} X_{b}^{i} + D(X_{-a}, w)$$

$$- \sum_{a,b} c \left( \sum_{k} \phi_{a-b}^{k} \frac{\lambda_{k}}{\lambda_{k1}} + (1 - \sum_{k} \phi_{a-b}^{k}) \frac{1}{\lambda} \right) |w_{ab}| + C. \quad (19)$$

Therefore we have the following proposition.

Proposition 5. The M step is equivalent to the following general Lasso-like optimization problem.

$$\hat{w} = \arg \max_{w} \sum_{i} \sum_{a,b} w_{ab} X_{a}^{i} X_{b}^{i} + D(X_{-a}, w)$$

$$- \sum_{a,b} \rho_{ab} |w_{ab}|, \quad (20)$$

where

$$\rho_{ab} = c \sum_{k} \phi_{a-b}^{k} \phi_{a-b}^{k} \frac{\lambda_{k}}{\lambda_{k1}} + (1 - \sum_{k} \phi_{a-b}^{k}) \frac{1}{\lambda} \epsilon \quad (21)$$

Note that $\rho$ incorporates all the information we need from hidden variables to compute the penalty term. Therefore we can treat it as a LASSO-like problem, which can be solved efficiently by the fast iterative shrinkage-thresholding algorithm (FISTA) [7] or other similar algorithms.

A Poisson Graphical Model Example

In this part we present an example to show how Proposition 5 works on an exponential family besides GGMs. For Poisson Graphical Models, the log conditional probability density function for node $a$ can be written as

$$\log P(X_{a}|X_{-a}, w_{a}) = \sum_{b \neq a} w_{ab} X_{b}$$

$$- \exp(w_{a} + \sum_{b \neq a} w_{ab} X_{b}) - \log(X_{a})$$

Therefore the optimization problem is

$$\max_{w} \frac{1}{n} \sum_{a,i} \left( \exp(w_{a} + \sum_{b \neq a} w_{ab} X_{b}) + X_{a}^{i} w_{a} + \sum_{b \neq a} w_{ab} X_{b}^{i} \right)$$

$$s.t. w_{ab} = w_{ba} \forall a, b. \quad (22)$$

The problem above can easily be optimized by any soft thresholding algorithms. Note that $w_{a}$ is a scalar indicating the constant term for linear interactions between node $a$ and the others.

We summarize our algorithm as follows.

Algorithm 2 Clustering Graphical Models

Input: $X_{n \times p}$

while $Q(w^{'t})$ does not converge do

//E Step by Stochastic Variational Inference

Compute $\gamma$, $\lambda$, $\phi$ by Algorithm 1

//M Step by Lasso-like algorithm

Compute $\rho$ by Eq. 21

Compute $w^{'t+1}$ by solving Eq. 20

$t \leftarrow t+1$

end while

4 Experimental Results

In this section we present some experimental results of our method and its comparison with a heuristic approach, i.e.
learn the graph by generalized linear model first, and then cluster the graph by MMSB, on synthetic data. Rand index and Hamming distance is applied for evaluation purpose. Rand index is a measure for the similarity of two clustering results. Rand index ranges from 0 to 1, with 1 indicating the exact match and 0 the worst. Hamming distance measures the distance between two graphs, with 0 indicating that two graphs are same. We further apply our method to a RNA data set to show that it can detect some biologically meaningful clusters.

4.1 Synthetic Data

Suppose we want to generate a network with \( p \) random variables and \( K \) clusters. We can use a simple version of MMSB model to generate the graph. In particular, we set \( \beta_k \) (the probability that there is an edge within block \( k \), i.e. cluster strength) to 0.2 for \( k = 1, \ldots, K \). We also assign \( \gamma_a \) to block \( k \) (\( \gamma_{ak} = 1 \)) with probability \( \frac{1}{K} \). Finally we compute \( r_{ab} \) according to the third step of our model for each pair of nodes \( a \) and \( b \), and sample an edge between \( a \) and \( b \) by probability \( r_{ab} \). Furthermore, we set \( \epsilon = 0.02 \), \( p = 128 \), and \( K = 4 \). Given the graph, we can then sample the data correspondingly.

To evaluate performance of our model more precisely, two kinds of graphical models in the exponential family are used: Gaussian Graphical Models (GGMs) and Poisson Graphical Models (PGMs). We use the heuristic approach (i.e., estimate the graph first before clustering it) as the baseline. Note that the expectation of estimated \( \pi_{ak} \) for each node \( a \) can be treated as the probability that node \( a \) is in block \( k \). To compare with the ground truth, we assign node \( a \) to block \( l \) such that \( \pi_{al} \) is maximized over all \( \pi_{ak} \), \( k = 1, \ldots, K \).

For GGMs, we set the precision matrix element \( \Omega_{ab} = 0.3 \) if there is an edge between node \( a \) and node \( b \), and 0 otherwise. To make sure \( \Omega \) is positive definite, we set its diagonal to the absolute of the minimum eigenvalue of \( \Omega \) plus 0.2. For PGMs, the conditional distribution for node \( a \) is

\[
P(X_a | X_{-a}, w_a) = \exp(X_a \sum_{b \neq a} w_{ab} X_b - \log(X_a!)) - A(w_a, X_{-a}) \tag{23}
\]

where \( \log(X_a!) \) is the base measure and \( A(w_a, X_{-a}) \) is log-normalizing constant. Some simple algebra can show that \( A(w_a, X_{-a}) = \exp(\sum_{b \neq a} w_{ab} X_b) \) and \( w_{ab} \leq 0 \) for all \( a, b \) so that \( A(w) < \infty \) [8]. Given the conditional distribution, Gibbs sampling is used to generate simulated data for PGMs.

We set \( p = 128 \) for both GGMs and PGMs. The simulation is conducted for each sample size \( N \), where
Figure 3: The estimated network from a set of breast cancer miRNA data. The five colors in the graph indicate different clusters, and bigger nodes imply the overlapping nodes. Most of the clusters are clear (Eg. the yellow, blue and green clusters).

\( N \in \{64, 96, 128, 192, 256\}, \) i.e. \( \{0.5, 0.75, 1, 1.5, 2\} \times p, \) for GGMs. For PGMs, we set sample size \( N \) to \( \{96, 128, 196, 256, 384\}, \) i.e. \( \{0.75, 1, 1.5, 2, 3\} \times p. \) The simulation results are illustrated in Figure 2. It shows that our algorithm outperforms the heuristic approach for nearly all sample sizes in terms of both clustering accuracy (Rand Index) and graph estimation accuracy (Hamming distance). As for running time, our EM algorithm is about two times slower than heuristic approach since we use warm start.

4.2 Real Data

To test the performance of our method, we apply it to a breast cancer microRNA (miRNA) data set from next generation sequencing data. The data set is obtained from Cancer Genome Atlas (TCGA) [21], and preprocessed according to the method in [2]. The final dataset has 416 variables and 452 samples. An EM Poisson graphical model with \( \epsilon = 0.05 \) is fitted to estimate and cluster the graph at the same time. Note that GGMs cannot be easily applied here because the data consist of counts. After removing cluster with size less than 5, the final resulting network with 374 miRNAs is illustrated in Figure 3, where the color indicates our clustering result. Bigger nodes indicates “overlapping” nodes that probably play the role of connecting two different blocks. Although we cannot evaluate our result based on Rand index or Hamming distance since the labels of cluster assignment as well as network are not available, there are still several interesting results from a biological perspective. For example, our algorithm identifies 8 overlapping nodes (i.e., miRNAs). Amongst them, it is known that HSA-MIR-146A [23], HSA-MIR-200B[4] and HSA-MIR-200C [4] play a very important role in identifying breast cancer gene targets. In contrast, other models such as GGMs or PGMs cannot identify such targets since only hard clustering is involved. Further, those overlapping node cannot be identified by simply selecting nodes with highest degree since their degrees are 10, 7, 7, 4, 8, 7, 7 and 5 respectively, while the 8-th highest degree of all nodes is 14.

Our clustering result is also consistent with some biological experimental results. For example, in the purple cluster, the non-coding miRNAs HSA-LET-7g, MIR-200C, HSA-MIR-181B-1 and HSA-MIR-181B are all associated with Chemoresponse to S-1 in Colon Cancer [20]. In the red cluster, the HSA-LET-7A family members are a modulator of KLK6 protein expression that is independent of the KLK6 copy number status. Further, the miRNAs which have been identified to have no direct relationship with KLK6 copy number status, such as HSA-MIR-296 and HSA-MIR-296, do not appear in the red cluster[6]. On the other hand, empirical approach fails to detect such information.

\[ \text{In this case, a node is called an overlapping node if it belongs to two blocks with probability larger than or equal to 0.4} \]
5 Conclusions

We present a generative model that can simultaneously detect the overlapping block structure and estimate the graph by applying an EM algorithm with variational inference. Experimental results show that our method outperforms the heuristic approach on both synthetic and real data. In particular, our algorithm not only applies to Gaussian Graphical models, but also to all kinds of models belonging to general exponential families, such as Poisson distribution and multinomial distribution. Besides, our method can obtain a soft clustering result and detect overlapping nodes due to the application of MMSB.

In future we will work on the derivation and improvement of our method in high dimensional data (i.e. $n \ll p$) settings, where it shall be more effective and helpful. The reason is that with enough data, one can always estimate graph first, then cluster it. Furthermore, theoretical results also need to be established for our method, since we do not have consistent guarantee as of yet.

Acknowledgements

This work is supported by the Alfred P. Sloan Research Fellowship to JX and NSF Career Award NSF/CCF AF-1149811 to JX.

References


