Latent Variable Fisher SBM

1 Approach

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The adjacency matrix $\mathbf{A}$ is defined such that $\mathbf{A}_{uv} = 1$ if $(u, v) \in \mathcal{E}$ and 0 otherwise. We assume that $\mathcal{G}$ is generated by a Stochastic Block Model (SBM), so that each node $u \in \mathcal{V}$ belongs to one of k communities.

Community membership is defined by the partition matrix $\mathbf{Z}$ such that $\mathbf{Z}_{ui} = 1$ if node u is a member of community i and 0 otherwise. The structure matrix $\mathbf{\Theta}$ is defined such that $\mathbf{\Theta}_{ij}$ is the expected number of edges (or edge weight) from nodes in community i to nodes and community j. That is, each element of the adjacency matrix is generated i.i.d. with conditional expectation $\mathrm{E}(\mathbf{A}_{uv}|\mathbf{Z}) = \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v$.

1.1 Bernoulli

In the simple definition of $\mathcal{G}$ (see above), the adjacency matrix $\mathbf{A}$ only takes elements in $(0, 1)$. Therefore, this type of graph is well-described by the assumption

$$\mathbf{A}_{uv}|\mathbf{Z} \sim \mathrm{Bernoulli}(\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v)\,.$$

The log-likelihood of this model is

$$\begin{aligned}
\ell(\mathbf{Z}, \mathbf{\Theta}|\mathbf{A}) &= \sum_{u,v} \big[\mathbf{A}_{uv} \ln\big(\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v\big) + (1-\mathbf{A}_{uv}) \ln\big(1-\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v\big)\big] && (1)\\
&= \sum_{u,v} [\mathbf{A}_{uv} \ln(\mathbf{P}_{uv}) + (1-\mathbf{A}_{uv}) \ln(1-\mathbf{P}_{uv})] && (2)
\end{aligned}$$

where $\mathbf{P}_{uv} = \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v$. The gradient of this log-likelihood is

$$\begin{aligned}
\frac{\partial \ell}{\partial \mathbf{Z}_u} &= \sum_v \left[\frac{\mathbf{A}_{uv} - \mathbf{P}_{uv}}{\mathbf{P}_{uv}(1-\mathbf{P}_{uv})}\right] \mathbf{\Theta} \mathbf{Z}_v && (3)\\
&= \mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} (\mathbf{A}_u - \mathbf{P}_u) && (4)
\end{aligned}$$

where

$$\mathbf{W}^{[u]} = \mathrm{diag}\left(\dots \frac{1}{\mathbf{P}_{uv}(1-\mathbf{P}_{uv})} \dots\right)\,. \quad (5)$$

Note that the weight matrix $\mathbf{W}^{[u]}$ is defined over nodes $v \in \mathcal{V}$ with fixed u. The hessian of this log-likelihood is

$$\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}'_u} = -\sum_v \left[\frac{\mathbf{A}_{uv} - \mathbf{P}_{uv}}{\mathbf{P}_{uv}(1-\mathbf{P}_{uv})}\right]^2 \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}'_v \mathbf{\Theta}' \tag{6}$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{S}^{[u]} \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \tag{7}$$

where

$$\mathbf{S}^{[u]} = \operatorname{diag}\big(\ldots (\mathbf{A}_{uv} - \mathbf{P}_{uv})^2 \ldots\big) \,. \tag{8}$$

Note that the squared-error matrix $\mathbf{S}^{[u]}$ is defined over nodes $v \in \mathcal{V}$ with fixed u. The expectation of the hessian is

$$\mathrm{E}\left[\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}'_u}\right] = -\sum_v \left[\frac{\operatorname{var}(\mathbf{A}_{uv})}{\mathbf{P}^2_{uv}(1-\mathbf{P}_{uv})^2}\right] \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}'_v \mathbf{\Theta}' \tag{9}$$

$$= -\sum_v \left[\frac{1}{\mathbf{P}_{uv}(1-\mathbf{P}_{uv})}\right] \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}'_v \mathbf{\Theta}'$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \,. \tag{10}$$

Note that $\mathrm{E}(\mathbf{A}_{uv} - \mathbf{P}_{uv})^2 = \operatorname{var}(\mathbf{A}_{uv}) = \mathbf{P}_{uv}(1-\mathbf{P}_{uv})$.

It should be noted that the gradients and hessians discussed here are inexact, as the case where $u = v$ requires an additional scaling factor. That is, $\frac{d\mathbf{P}_{uv}}{d\mathbf{Z}_u} = \mathbf{\Theta}\mathbf{Z}_v$ if $u \neq v$ and $\frac{d\mathbf{P}_{uv}}{d\mathbf{Z}_u} = 2\mathbf{\Theta}\mathbf{Z}_v$ if $u = v$. This factor is ignored for ease of notation.

1.2 Poisson

Now consider the multi-graph definition of $\mathcal{G}$. Here, let $\mathbf{A}_{uv} \in \mathbb{N}_0$ be the number of parallel edges from node u to node v. This type of graph is well-described by the assumption

$$\mathbf{A}_{uv} | \mathbf{Z} \sim \operatorname{Poisson}(\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v) \,.$$

The log-likelihood of this model is

$$\ell(\mathbf{Z}, \mathbf{\Theta} | \mathbf{A}) = \sum_{u,v} \left[\mathbf{A}_{uv} \ln\big(\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v\big) - \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v - \ln(\mathbf{A}_{uv}!)\right] \tag{11}$$

$$\propto \sum_{u,v} \left[\mathbf{A}_{uv} \ln(\mathbf{P}_{uv}) - \mathbf{P}_{uv}\right] \tag{12}$$

where again $\mathbf{P}_{uv} = \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v$. The gradient of this log-likelihood is

$$\frac{\partial \ell}{\partial \mathbf{Z}_u} = \sum_v \left[\frac{\mathbf{A}_{uv} - \mathbf{P}_{uv}}{\mathbf{P}_{uv}}\right] \mathbf{\Theta} \mathbf{Z}_v \tag{13}$$

$$= \mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} (\mathbf{A}_u - \mathbf{P}_u) \tag{14}$$

where

$$\mathbf{W}^{[u]} = \text{diag}\left(\dots \frac{1}{\mathbf{P}_{uv}} \dots\right) . \tag{15}$$

Note that this weight matrix $\mathbf{W}^{[u]}$ is again defined over nodes $v \in \mathcal{V}$ with fixed u. The hessian of this log-likelihood is

$$\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}'_u} = -\sum_v \left[\frac{\mathbf{A}_{uv}}{\mathbf{P}^2_{uv}}\right] \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}'_v \mathbf{\Theta}' \tag{16}$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Y}^{[u]} \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \tag{17}$$

where

$$\mathbf{Y}^{[u]} = \text{diag}(\mathbf{A}_u) . \tag{18}$$

The expectation of the hessian is

$$\mathrm{E}\left[\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}'_u}\right] = -\sum_v \left[\frac{1}{\mathbf{P}_{uv}}\right] \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}'_v \mathbf{\Theta}' \tag{19}$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \tag{20}$$

Note that $\mathrm{E}(\mathbf{A}_{uv}) = \mathbf{P}_{uv}$.

1.3 Normal

Now consider a graph $\mathcal{G}$ defined with real-valued edge weights. Here, let $\mathbf{A}_{uv} \in \mathbb{R}$ be the weight of an edge from node u to node v. This type of graph can be described by the assumption

$$\mathbf{A}_{uv} | \mathbf{Z} \sim \text{Normal}(\mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v,\, \sigma^2) .$$

The log-likelihood of this model is

$$\ell(\mathbf{Z}, \mathbf{\Theta} | \mathbf{A}) = \sum_{u,v} \left[\ln\left(\frac{1}{\sqrt{2\pi}\sigma}\right) + \frac{1}{2\sigma^2} (\mathbf{A}_{uv} - \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v)^2\right] \tag{21}$$

$$\propto \frac{1}{2} \sum_{u,v} (\mathbf{A}_{uv} - \mathbf{P}_{uv})^2 \tag{22}$$

where we let $\sigma^2 = 1$ and once again $\mathbf{P}_{uv} = \mathbf{Z}'_u \mathbf{\Theta} \mathbf{Z}_v$. The gradient of this log-likelihood is simply

$$\frac{\partial \ell}{\partial \mathbf{Z}_u} = \sum_v \left[\mathbf{A}_{uv} - \mathbf{P}_{uv}\right] \mathbf{\Theta} \mathbf{Z}_v \tag{23}$$

$$= \mathbf{\Theta} \mathbf{Z}' (\mathbf{A}_u - \mathbf{P}_u) . \tag{24}$$

The hessian of this log-likelihood is

$$\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}_u'} = -\sum_v \mathbf{\Theta} \mathbf{Z}_v \mathbf{Z}_v' \mathbf{\Theta}' \tag{25}$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{Z} \mathbf{\Theta}' \tag{26}$$

$$= -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \tag{27}$$

where

$$\mathbf{W}^{[u]} = \mathbf{I} \tag{28}$$

is constant for all nodes. The expectation of the hessian is

$$\mathrm{E}\left[\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}_u'}\right] = -\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}' \,. \tag{29}$$

1.4 MLEs

To estimate the pararmeters any of the above SBMs, we use the closed form maximum likelihood estimate (MLE) for $\mathbf{\Theta}$ and a modification of the Fisher scoring method for $\mathbf{Z}$. It is convenient that the MLE of $\mathbf{\Theta}_{ij}$ for all three models is simply the average over the adjacency matrix between nodes in community i and community j. Specifically,

$$\left[\hat{\mathbf{\Theta}}_{\text{MLE}}(\mathbf{A}, \mathbf{Z})\right]_{ij} = \frac{\mathbf{M}_{ij}}{\mathbf{n}_i \mathbf{n}_j} \tag{30}$$

where

$$\mathbf{M} = \mathbf{Z}' \mathbf{A} \mathbf{Z} \,, \qquad \mathbf{n} = \sum_u \mathbf{Z}_u \,. \tag{31}$$

1.5 Fisher Update

In the standard Fisher scoring method, the update rule for each row of $\mathbf{Z}$ is

$$\mathbf{Z}_u \leftarrow \mathbf{Z}_u - \mathrm{E}\left[\frac{\partial \ell^2}{\partial \mathbf{Z}_u \partial \mathbf{Z}_u'}\right]^{-1} \left[\frac{\partial \ell}{\partial \mathbf{Z}_u}\right] \tag{32}$$

$$= \mathbf{Z}_u + \left(\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}'\right)^{-1} \mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} (\mathbf{A}_u - \mathbf{P}_u) \tag{33}$$

$$= \left(\mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{Z} \mathbf{\Theta}'\right)^{-1} \mathbf{\Theta} \mathbf{Z}' \mathbf{W}^{[u]} \mathbf{A}_u \tag{34}$$

where $\mathbf{P}_{uv} = \mathbf{Z}_u' \mathbf{\Theta} \mathbf{Z}_v$ and $\mathbf{W}^{[u]}$ is given by equation 5, 15, or 28 depending on assumptions of the underlying graph.

This approach works well when the variable in question is continuous and unbounded. In our case however, $\mathbf{Z}_u$ is a restricted to a one-hot encoded vector. For this reason, we introduce a continuous, unconstrained latent variable $\mathbf{X}_u \in \mathbb{R}^k$ and let

$$\mathbf{Z}_u = \mathrm{hardmax}(\mathbf{X}_u) \qquad \forall\, u \in \mathcal{V} \tag{35}$$

where hardmax($\mathbf{x}$) is the one-hot encoding of $\arg\max_i \mathbf{x}_i$. The new update rule is applied to $\mathbf{X}$ by

$$\mathbf{X}_u \leftarrow \mathbf{X}_u + \left(\mathbf{\Theta}\mathbf{Z}'\mathbf{W}^{[u]}\mathbf{Z}\mathbf{\Theta}'\right)^{-1}\mathbf{\Theta}\mathbf{Z}'\mathbf{W}^{[u]}(\mathbf{A}_u - \mathbf{P}_u) \tag{36}$$

$$= (\mathbf{X}_u - \mathbf{Z}_u) + \left(\mathbf{\Theta}\mathbf{Z}'\mathbf{W}^{[u]}\mathbf{Z}\mathbf{\Theta}'\right)^{-1}\mathbf{\Theta}\mathbf{Z}'\mathbf{W}^{[u]}\mathbf{A}_u \tag{37}$$

In this way, the latent variable $\mathbf{X}$ is iteratively updated using the gradient and hessian computed with respect to its projection $\mathbf{Z}$.

To avoid iterating over each node $u \in \mathcal{V}$, it is convenient to update the full updates all at once instead of individually for every row. Thus we define the average weight matrix

$$\overline{\mathbf{W}} = \frac{1}{|\mathcal{V}|}\sum_u \mathbf{W}^{[u]} \tag{38}$$

and apply the update rule

$$\mathbf{X} \leftarrow (\mathbf{X} - \mathbf{Z}) + \left[\left(\mathbf{\Theta}\mathbf{Z}'\overline{\mathbf{W}}\mathbf{Z}\mathbf{\Theta}'\right)^{-1}\mathbf{\Theta}\mathbf{Z}'\overline{\mathbf{W}}\mathbf{A}\right]'. \tag{39}$$

This offers a significant increase in computational efficiency. To ensure stability and smoothness, we also add a regularization term $\alpha\mathbf{I}$ to the hessian, where α is the regularizer strength.

The MLE for $\mathbf{\Theta}$ is re-computed before each update. The full algorithm iterates over the following three steps until convergence is reached:

$$\textit{i.)} \qquad \mathbf{Z} \leftarrow \text{hardmax}(\mathbf{X}) \tag{40}$$

$$\textit{ii.)} \qquad \mathbf{\Theta} \leftarrow \hat{\mathbf{\Theta}}_{\text{MLE}}(\mathbf{A}, \mathbf{Z}) \tag{41}$$

$$\textit{iii.)} \qquad \mathbf{X} \leftarrow (\mathbf{X} - \mathbf{Z}) + \left[\left(\mathbf{\Theta}\mathbf{Z}'\overline{\mathbf{W}}\mathbf{Z}\mathbf{\Theta}' + \alpha\mathbf{I}\right)^{-1}\mathbf{\Theta}\mathbf{Z}'\overline{\mathbf{W}}\mathbf{A}\right]' \tag{42}$$

We initialize $\mathbf{X}$ randomly and close to $1/k$. Convergence is reached when $\mathbf{Z}$ changes by less than ϵ after each update.

2 Next Steps

- Further gains in computational efficiency might be achieved through diagonalization of the hessian matrix
- It should be simple to incorporate degree-corrected SBMs
- A sparsity parameter can also probably be incorporated
- Try using a similarity/distance matrix to stand in for a real-valued adjacency matrix and then apply the normal assumption a tabular clustering approach
- Is there a way to dynamically identify k? Maybe $L1$ regularization?
- Recursive least squares? Dynamic graph setting