Latent Networks Models

Game of Thrones

Lily Chou, Ben Draves, Nathan Josephs, Kelly Kung

Department of Mathematics and Statistics
Boston University

December 9, 2018

Overview

Latent Network Model

Using Hoff's work, we model the presence of an edge given our latent variables as

$$\text{logit}\ \mathbb{P}(Y_{ij} = 1|Z) = ||Z_i - Z_j|| + \epsilon_{ij}$$

where $||Z_i - Z_j||$ is the latent distance between nodes i and j and

$$Z_i \overset{ind}{\sim} \sum_{k=1}^{G} \lambda_k \text{MVN}_d(\mu_k, \sigma_k^2 I_d)$$

Latent Network Model: Priors

$$
\begin{aligned}
Y_{ij}|Z_i, Z_j &\overset{ind}{\sim} \text{Bern}\left[\text{logit}^{-1}(\|Z_i - Z_j\|)\right] \\
Z_i|K_i = k_i &\overset{ind}{\sim} MVN(\mu_{k_i}, \sigma^2_{k_i} I_d) \\
K &\overset{iid}{\sim} \text{Multinoulli}(G, \lambda) \\
\lambda_k &\overset{iid}{\sim} \frac{1}{G} \\
\mu_k &\overset{iid}{\sim} \text{MVN}_d(0, I_2) \\
\sigma^2_k &\overset{iid}{\sim} \text{Inv}\chi^2_1
\end{aligned}
$$

Latent Network Model: Likelihood

$$
\begin{aligned}
\mathcal{L}(Z,\theta;Y) &= \prod_{i<j} \mathbb{P}(Y_{ij}|Z_i,Z_j)\mathbb{P}(Z_i|K_i,\mu_{k_i},\sigma^2_{k_i})\mathbb{P}(Z_j|K_j,\mu_{k_j},\sigma^2_j) \\
&\qquad \times \mathbb{P}(K_i|\lambda_i)\mathbb{P}(\lambda_i)\mathbb{P}(\mu_{k_i})\mathbb{P}(\sigma^2_{k_i})\mathbb{P}(K_j)\mathbb{P}(\mu_{k_j})\mathbb{P}(\sigma^2_{k_j}) \\
&\propto \prod_{i<j} \Big(\text{logit}^{-1}(\|Z_i - Z_j\|)\Big)^{Y_{ij}}\Big(1-\text{logit}^{-1}(\|Z_i - Z_j\|)\Big)^{1-Y_{ij}} \\
&\times \frac{1}{(\sigma^2_{k_i})^{1/2}} \exp\Big\{-\frac{1}{2\sigma^2_{k_i}}(Z_i-\mu_{k_i})^T(Z_i-\mu_{k_i})\Big\}\ \frac{1}{(\sigma^2_{k_j})^{1/2}} \\
&\times \exp\Big\{-\frac{1}{2\sigma^2_{k_j}}(Z_j-\mu_{k_j})^T(Z_j-\mu_{k_j})\Big\}\exp\Big\{-\frac{1}{2}\mu_{k_i}^T\mu_{k_i}\Big\} \\
&\times \exp\Big\{-\frac{1}{2}\mu_{k_j}^T\mu_{k_j}\Big\} \times \frac{1}{(\sigma^2_{k_i})^2}\exp\Big\{-\frac{1}{\sigma^2_{k_i}}\Big\}\frac{1}{(\sigma^2_{k_j})^2}\exp\Big\{-\frac{1}{\sigma^2_{k_j}}\Big\} \\
&\times \lambda_i \times \lambda_j
\end{aligned}
$$

Unweighted Network Model

Let Y_{ij} indicate whether there is an edge E_{ij} between nodes i and j.

$$Y_{ij}|p_{ij} \overset{ind}{\sim} \text{Bern}(p_{ij})$$
$$p_{ij} \overset{iid}{\sim} \text{Beta}(\alpha, \beta)$$

where $p_{ij} \equiv 2 - 2 * \text{logit}^{-1}(d_{ij})$.

Then the log-likelihood for this model can be written as

$$l(p, \alpha, \beta; Y) = \sum_{i<j} Y_{ij} \log\left(\frac{p_{ij}}{1 - p_{ij}}\right) + \log(1 - p_{ij}) + \log\Gamma(\alpha + \beta) - \log\Gamma(\alpha) - \log\Gamma(\beta) + (\alpha - 1)\log p_{ij} + (\beta - 1)\log(1 - p_{ij})$$

Unweighted Network Model: E-Step

We can take the expectation of the log-likelihood given the data $\mathbf{Y}$ and parameters $\theta = (\alpha, \beta)$

$$
\begin{aligned}
Q(\theta;\theta^{(t)}) = \sum_{i<j} &(Y_{ij} + \alpha - 1)\mathbb{E}_{p_{ij}|Y_{ij},\theta^{(t)}}[\log p_{ij}] \\
&+ (\beta - Y_{ij})\mathbb{E}_{p_{ij}|Y_{ij},\theta^{(t)}}\big[\log(1 - p_{ij})\big] + \log\Gamma(\alpha+\beta) \\
&- \log\Gamma(\alpha) - \log\Gamma(\beta)
\end{aligned}
$$

Since $p_{ij}|Y_{ij},\theta \propto Beta(\alpha + Y_{ij}, \beta + 1 - Y_{ij})$, we define the following

$$
\pi_{ij} \equiv \mathbb{E}_{p_{ij}|Y_{ij},\theta}\big[\log p_{ij}\big] = \Psi\Big(\alpha + Y_{ij}\Big) - \Psi\Big(\alpha + \beta + 1\Big)
$$

$$
\eta_{ij} \equiv \mathbb{E}_{p_{ij}|Y_{ij},\theta}\big[\log(1 - p_{ij})\big] = \Psi\Big(\beta + 1 - Y_{ij}\Big) - \Psi\Big(\alpha + \beta + 1\Big)
$$

Unweighted Network Model: M-Step

To maximize, $Q(\theta, \theta^{(t)})$ with respect to θ, we see that $\theta^{(t+1)}$ must satisfy the following:

$$\Psi\Big(\alpha^{(t+1)} + \beta^{(t)}\Big) - \Psi(\alpha^{(t+1)}) = -\frac{\sum_{i<n} \mathbb{E}_{p_{ij}|Y_{ij},\theta^{(t)}}\big[\log p_{ij}\big]}{\binom{n}{2}} \qquad (\alpha_U)$$

$$\Psi\Big(\alpha^{(t+1)} + \beta^{(t+1)}\Big) - \Psi(\beta^{(t+1)}) = -\frac{\sum_{i<n} \mathbb{E}_{p_{ij}|Y_{ij},\theta^{(t)}}\big[\log(1-p_{ij})\big]}{\binom{n}{2}} \qquad (\beta_U)$$

Here, we use Newton-Raphson to obtain both $\alpha^{(t+1)}$ and $\beta^{(t+1)}$.

Unweighted Network Model: Pseudocode

Algorithm 1: EM for simplified latent network unweighted model

LNM EM (G, tol);

Input : Graph G

Tolerance tol

Output: Nuisance Parameters α^*, β^*

Latent Probability Estimates $\hat{p}$

Latent Distance Estimates $\hat{d}$

Initialize $Q^{(0)}$ **repeat**

  E: calculate $\pi^{(t)}$, $\eta^{(t)}$;

  M: update $\alpha^{(t+1)}$ using (α_U);

  update $\beta^{(t+1)}$ using (β_U);

  calculate $Q(\theta, \theta^{(t+1)})$

until $\left|\frac{Q(\theta^{(t+1)},\theta^{(t)})-Q(\theta^{(t)},\theta^{(t)})}{Q(\theta^{(t)},\theta^{(t)})}\right| < tol$;

return α^*, β^*, $\hat{p} = e^{\pi^*}$, $\hat{d} = logit^{-1}(1 - \frac{e^{\pi^*}}{2})$; where α^*, β^*, π^* are converged values

Unweighted Network Model: Heat Maps

Weighted Network Model

Let Y_{ij} be the weight on edge $E_{ij} \in \mathbf{E}$.

$$Y_{ij}|\lambda_{ij} \overset{ind}{\sim} \text{Pois}(\lambda_{ij})$$
$$\lambda_{ij} \overset{iid}{\sim} \text{Gamma}(\alpha, \beta)$$

Then the log-likelihood for this model can be written as

$$l(\lambda, \alpha, \beta; Y) = \sum_{i<j} \Big\{ \log \lambda_{ij}(Y_{ij} + \alpha - 1) - \lambda_{ij}(1 + \beta)$$
$$- \log(Y_{ij}!) + \alpha \log(\beta) - \log \Gamma(\alpha) \Big\}$$

Weighted Network Model: E-Step

Taking an expectation of this log-likelihood given the data $\mathbf{Y}$ and parameters $\theta = (\alpha, \beta)$

$$Q(\theta; \theta^{(t)}) = \sum_{i<j} \Big\{ (Y_{ij} + \alpha - 1)\mathbb{E}_{\lambda_{ij}|Y_{ij},\theta^{(t)}}\big[\log \lambda_{ij}\big] - (1+\beta)\mathbb{E}_{\lambda_{ij}|Y_{ij},\theta^{(t)}}\big[\lambda_{ij}\big] - \log(Y_{ij}!) + \alpha \log(\beta) - \log \Gamma(\alpha) \Big\}$$

Seeing as $\lambda_{ij}|Y_{ij}, \theta \propto \text{Gamma}(\alpha + Y_{ij}, \beta + 1)$ we can define

$$\pi_{ij} \equiv \mathbb{E}_{\lambda_{ij}|Y_{ij},\theta}\big[\lambda_{ij}\big] = \frac{\alpha + Y_{ij}}{1+\beta}$$

$$\eta_{ij} \equiv \mathbb{E}_{\lambda_{ij}|Y_{ij},\theta}\big[\log \lambda_{ij}\big] = \log(1+\beta) + \Psi(\alpha + Y_{ij})$$

Weighted Network Model: M-Step

Maximizing $Q(\theta, \theta^{(t)})$ with respect to θ, we see that $\theta^{(t+1)}$ must satisfy

$$\beta^{(t+1)} = \frac{\binom{n}{2}}{\sum_{i<j} \pi_{ij}} \alpha^{(t+1)}$$

$$\Psi(\alpha^{(t+1)}) = \frac{\sum_{i<j} \eta_{ij} + \binom{n}{2} \log(\beta^{(t+1)})}{\binom{n}{2}}$$

We first update $\beta^{(t+1)}$ using $\alpha^{(t)}$ then use Netwon-Raphson to attain $\theta^{(t+1)}$.

Weighted Network Model: Psuedo-Code

Algorithm 2: EM for simplified latent network weighted model

```
LNM EM (G, tol);
Input  : Graph G
         Tolerance tol
Output: Nuisance Parameters α*, β*
         Latent Mean Estimates λ̂
         Latent Distance Estimates d̂
Initialize Q^(0) repeat
    E: calculate π^(t), η^(t);
    M: update β^(t+1) using (β_W);
    update α^(t+1) using (α_W);
    calculate Q(θ, θ^(t+1))
until |(Q(θ^(t+1), θ^(t)) − Q(θ^(t), θ^(t))) / Q(θ^(t), θ^(t))| < tol;
return α*, β*, λ̂ = π*, d̂ = 1/π*; where α*, β*, π* are converged values
```

Weighted Network Model: Distance Estimates

Weighted Network Model: Distance Density

Weighted Network Model: λ Estimates

Weighted Network Model: Spectral Clustering