Latent Networks Models

Game of Thrones

Lily Chou, Ben Draves, Nathan Josephs, Kelly Kung[1]

[1]Department of Mathematics and Statistics
Boston University

December 9, 2018

Overview

Weighted Network Model

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

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

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

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

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 1: EM for simplified latent network weighted model

LNM EM (G, tol);

Input : Graph G
Tolerance tol

Output: Nuisance Parameters α^*, β^*
Latent Mean Estimates $\hat{\lambda}$
Latent Distance Estimates $\hat{d}$

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

- **E:** calculate $\pi^{(t)}$, $\eta^{(t)}$;
- **M:** update $\beta^{(t+1)}$ using (β_W);
- update $\alpha^{(t+1)}$ using (α_W);
- 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^*, $\hat{\lambda} = \pi^*$, $\hat{d} = \frac{1}{\pi^*}$; where α^*, β^*, π^* are converged values