

Latent Networks Models

Game of Thrones

Lily Chou, Ben Draves, Nathan Josephs, Kelly Kung

¹Department of Mathematics and Statistics
Boston University

December 9, 2018

Overview

- 1 Introduction
- 2 Latent Network Model
- 3 Expectation Maximization
 - Unweighted Network Model
 - Weighted Network Model

Weighted Network Model

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

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

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

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

Weighted Network Model: E-Step

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

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

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

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

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

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 Newton-Raphson to attain $\theta^{(t+1)}$.

Weighted Network Model: Psuedo-Code

Algorithm 1: EM for simplified latent network weighted model

1 LNM EM (G, tol);

Input : Graph G

Tolerance tol

Output: Nuisance Parameters α^*, β^*

Latent Mean Estimates $\hat{\lambda}$

Latent Distance Estimates \hat{d}

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

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

4 **M:** update $\beta^{(t+1)}$ using (β_W) ;

5 update $\alpha^{(t+1)}$ using (α_W) ;

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

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

8 **return** $\alpha^*, \beta^*, \hat{\lambda} = \pi^*, \hat{d} = \frac{1}{\pi^*}$; 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