

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

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

Data

- 1 Data origin: A Song of Ice and Fire · A storm of Swords
- 2 Data form: 352 pairs of characters and the number of interaction between them

We start our analysis by constructing a Weighted Network G :

- i Characters \rightarrow nodes $N_V(G) = 107$
- ii Interactions \rightarrow edges $N_E(G) = 352$
- iii numbers of interaction \rightarrow Weights on the edge

Form a Network

PICTURE HERE

- a Q: How to handle the sparsity of the network?
- b A: Take a subnetwork of G , cut off at 100 interactions, call it G'
- c $N_V(G') = 24$, $N_E(G') = 102$
- d Use adjacency matrix A to represent G' :
 - i $a(i, j)$ = number of interactions between i th and j th character
 - ii i.e., $a(i, j) = 0$ if there's no interaction between the two people

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

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

Latent Network Model: Priors

$$Y_{ij}|Z_i, Z_j \stackrel{ind}{\sim} \text{Bern}\left[\text{logit}^{-1}(\|Z_i - Z_j\|)\right]$$

$$Z_i|K_i = k_i \stackrel{ind}{\sim} \text{MVN}(\mu_{k_i}, \sigma_{k_i}^2 I_d)$$

$$K \stackrel{iid}{\sim} \text{Multinoulli}(G, \lambda)$$

$$\lambda_k \stackrel{iid}{\sim} \frac{1}{G}$$

$$\mu_k \stackrel{iid}{\sim} \text{MVN}_d(0, I_2)$$

$$\sigma_k^2 \stackrel{iid}{\sim} \text{Inv}\chi_1^2$$

Latent Network Model: Likelihood

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

Unweighted Network Model

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

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

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 $\theta = (\alpha, \beta)$

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

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

$$\begin{aligned} \pi_{ij} &\equiv \mathbb{E}_{p_{ij} | Y_{ij}, \theta} [\log p_{ij}] = \Psi(\alpha + Y_{ij}) - \Psi(\alpha + \beta + 1) \\ \eta_{ij} &\equiv \mathbb{E}_{p_{ij} | Y_{ij}, \theta} [\log(1 - p_{ij})] = \Psi(\beta + 1 - Y_{ij}) - \Psi(\alpha + \beta + 1) \end{aligned}$$

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:

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

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

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

1 **LNM EM** (G, tol);

Input : Graph G

Tolerance tol

Output: Nuisance Parameters α^*, β^*

Latent Probability Estimates \hat{p}

Latent Distance Estimates \hat{d}

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

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

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

5 update $\beta^{(t+1)}$ using (β_U) ;

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

Unweighted Network Model: Distances

Unweighted Network Model: Probabilities

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 2: 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