

Latent Networks Models

Game of Thrones

Lily Chou, Ben Draves, Nathan Josephs, Kelly Kung

Department of Mathematics and Statistics
Boston University

December 10, 2018

Overview

- 1 Introduction
- 2 Latent Network Model
- 3 Expectation Maximization
 - Unweighted Network Model
 - Weighted Network Model
- 4 Markov Chain Monte Carlo
- 5 Model Comparison
- 6 Conclusion

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 interactions 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 Number of interactions \rightarrow Weights on the edge

Form a Network

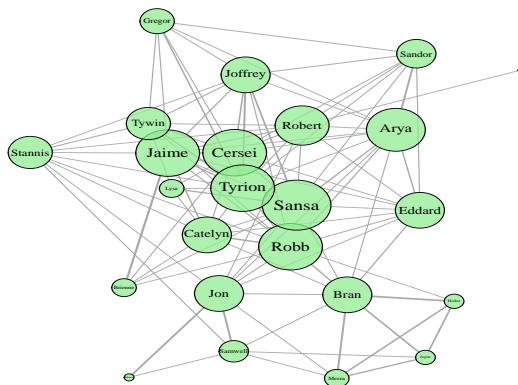
- a Q: How to handle the sparsity of the network?
- b A: Take a subnetwork of G , cut off at 100 interactions, call it G'
- c $N_V(G') = 24$,
 $N_E(G') = 102$
- d Use adjacency matrix A to represent G' :
 - i $a(i, j)$ = number of interactions between i th and j th character
 - ii i.e., $a(i, j) = 0$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

Unfiltered Network



Filtered Network

Filtered Network



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

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

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

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

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

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

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

Latent Network Model: Likelihood

$$\begin{aligned}\mathcal{L}(Z, \theta; Y) = & \prod_{i < j} \mathbb{P}(Y_{ij} | Z_i, Z_j) \mathbb{P}(Z_i | K_i, \mu_{k_i}, \sigma_{k_i}^2) \mathbb{P}(Z_j | K_j, \mu_{k_j}, \sigma_j^2) \\ & \times \mathbb{P}(K_i | \lambda_i) \mathbb{P}(\lambda_i) \mathbb{P}(\mu_{k_i}) \mathbb{P}(\sigma_{k_i}^2) \mathbb{P}(K_j) \mathbb{P}(\mu_{k_j}) \mathbb{P}(\sigma_{k_j}^2)\end{aligned}$$

Latent Network Model: Likelihood

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

Unweighted Network Model

Let Y_{ij} be the random variable that indicates whether there is an edge E_{ij} between nodes i and j .

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

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

Unweighted Network Model

Let Y_{ij} be the random variable that indicates whether there is an edge E_{ij} between nodes i and j .

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

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

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

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

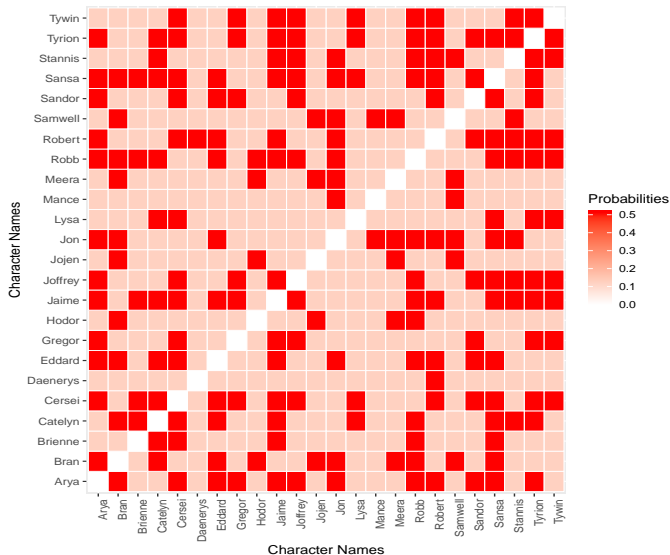
7 **until** $\left| \frac{Q(\theta^{(t+1)}, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})}{Q(\theta^{(t)}, \theta^{(t)})} \right| < tol$;

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 $E_{ij} \in \mathbf{E}$.

$$Y_{ij} | \lambda_{ij} \stackrel{\text{ind}}{\sim} \text{Pois}(\lambda_{ij})$$

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

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

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

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

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

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

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

Weighted Network Model: Pseudocode

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

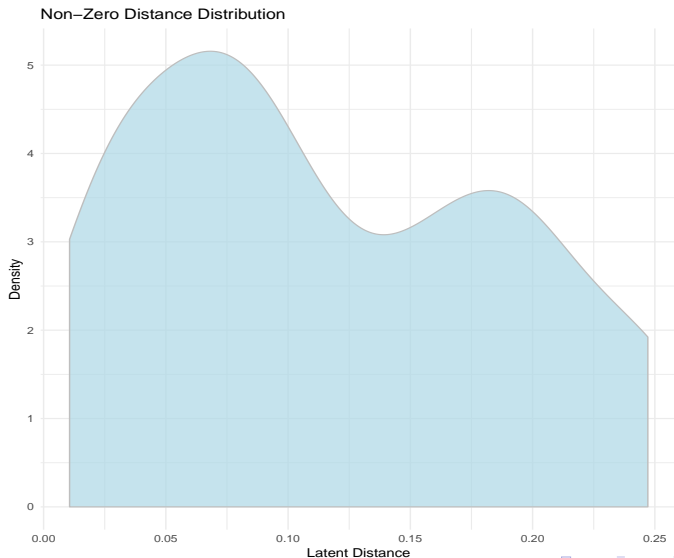
7 **until** $\left| \frac{Q(\theta^{(t+1)}, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})}{Q(\theta^{(t)}, \theta^{(t)})} \right| < tol$;

8 **return** $\alpha^*, \beta^*, \hat{\lambda} = \pi^*, \hat{d} = \frac{1}{\pi^*}$; where α^*, β^*, π^* are converged values

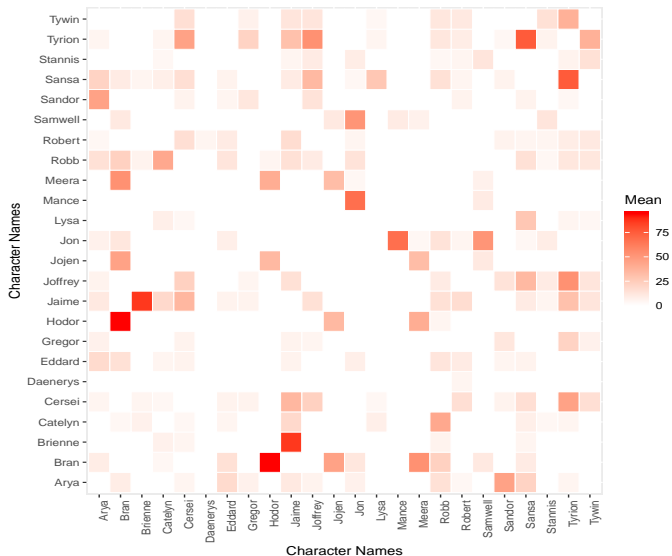
Weighted Network Model: Distance Estimates



Weighted Network Model: Distance Density



Weighted Network Model: λ Estimates



Weighted Network Model: Inference

- 1 Now that we have estimates for $\hat{\lambda}_{ij}$, how can we use these estimates to infer communities in the network?
- 2 One idea: Spectral Clustering

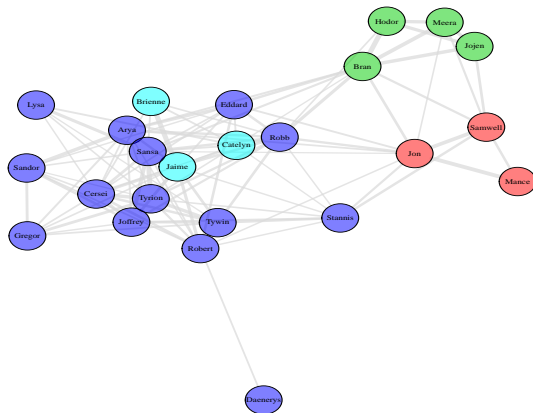
Weighted Network Model: Inference

- 1 Now that we have estimates for $\hat{\lambda}_{ij}$, how can we use these estimates to infer communities in the network?
- 2 One idea: Spectral Clustering
- 3 Interpret these estimates as smooth estimates of the weighted - Adjacency matrix

$$\hat{\Lambda} = \begin{bmatrix} \hat{\lambda}_{11} & \hat{\lambda}_{12} & \dots & \hat{\lambda}_{1n} \\ \hat{\lambda}_{21} & \hat{\lambda}_{22} & \dots & \hat{\lambda}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\lambda}_{n1} & \hat{\lambda}_{n2} & \dots & \hat{\lambda}_{nn} \end{bmatrix}$$

- 4 Spectral Clustering Visualization

Weighted Network Model: Spectral Clustering



Latent Network Model

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

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

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

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

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

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

Conditionals: μ

$$\begin{aligned}
 f_{\mu_k|\theta^{(t)}, Y}(\mu_k|\theta^{(t)}, Y) &\propto \prod_{k_i=k} \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} (Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) \right\} \exp \left\{ -\frac{1}{2} \mu_{k_i}^T \mu_{k_i} \right\} \\
 &\propto \exp \left\{ \sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\} \left[-\frac{(\sigma_{k_i}^2 + 1)}{2\sigma_{k_i}^t} \left(\mu_{k_i} - \frac{Z_i}{(\sigma_{k_i}^2 + 1)} \right)^T \left(\mu_{k_i} - \frac{Z_i}{(\sigma_{k_i}^2 + 1)} \right) \right] \right\}
 \end{aligned}$$

Thus for all $k \in \{1, \dots, G\}$, we arrive at the following distribution for $\mu_k|\theta^{(t)}, Y$:

$$\mu_k|\theta^{(t)}, Y \sim f_{MVN_d} \left(\sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\} \frac{Z_i^{(t)}}{(\sigma_{k_i}^2)^{(t)} + 1}, \sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\} \frac{(\sigma_{k_i}^2)^{(t)}}{(\sigma_{k_i}^2)^{(t)} + 1} I_2 \right)$$

Conditionals: σ^2

$$\begin{aligned}
 f_{\sigma_{k_i}^2 | \theta, Y}(\sigma_{k_i}^2 | \theta, Y) &\propto \prod_{K_i=k} \sigma_{k_i}^2^{-\frac{1}{2}} \exp \left\{ \frac{1}{2\sigma_{k_i}^2} (Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) \right\} (\sigma_{k_i}^2)^{-\frac{\epsilon}{2}-1} \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} \right\} \\
 &\propto (\sigma_{k_i}^2)^{\left(\left(\frac{-\epsilon-1}{2} ng - ng + 1 \right) - 1 \right)} \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} \sum_{K_i=k} \left((Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) + 1 \right) \right\}
 \end{aligned}$$

Conditionals: σ^2

$$f_{\sigma_{k_i}^2 | \theta, Y}(\sigma_{k_i}^2 | \theta, Y) \propto \prod_{K_i=k} \sigma_{k_i}^{2-\frac{1}{2}} \exp \left\{ \frac{1}{2\sigma_{k_i}^2} (Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) \right\} (\sigma_{k_i}^2)^{-\frac{c}{2}-1} \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} \right\} \\ \propto (\sigma_{k_i}^2)^{\left(\left(\frac{c-1}{2} n_g - n_g + 1 \right) - 1 \right)} \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} \sum_{K_i=k} \left((Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) + 1 \right) \right\}$$

Thus for all $k \in \{1, \dots, G\}$, we arrive at the following distribution for $\sigma_k^2 | \theta^{(t)}, Y$:

$$\sigma_k^2 | \theta^{(t)}, Y \sim \text{Inv}\Gamma \left(\frac{c}{2}, \frac{1}{2} \right) \stackrel{D}{=} \tau^2 \nu \text{Inv}\chi_c^2$$

where $n_g = \sum \mathbb{I}_{\{k_i=K\}}$ and $SS_g + n_g = \sum_{K_i=k} \left((Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) + 1 \right)$ and

$$\nu_{\text{post}} = (c+1)n_g + 2(n_g - 1) \\ \tau_{\text{post}}^2 = \frac{SS_g + n_g}{(c+1)n_g + 2(n_g - 1)},$$

Conditionals: Group K

$$\mathbb{P}(K_i = k | \theta, Y) \propto \lambda_k f_{MVN_d(\mu_k, \sigma_k^2)}(Z_i)$$

Since K is Multinoulli, we arrive at the following probability by recognizing they must normalize to unity:

$$\begin{aligned} \mathbb{P}(K_i = k | \theta, Y) &= \frac{\lambda_k f_{MVN_d(\mu_k, \sigma_k^2)}(Z_i)}{\sum_{g=1}^G \lambda_g f_{MVN_d(\mu_g, \sigma_g^2)}(Z_i)} \\ &= \frac{f_{MVN_d(\mu_k, \sigma_k^2)}(Z_i)}{\sum_{g=1}^G f_{MVN_d(\mu_g, \sigma_g^2)}(Z_i)} \quad (\lambda^{(t)}) \end{aligned}$$

Conditionals: Latent variable Z

$$f_{Z_i|\theta,Y}(Z_i|\theta,Y) \propto \prod_{j \neq i} \left(\text{logit}^{-1}(\|Z_i - Z_j\|) \right)^{Y_{ij}} \left(1 - \text{logit}^{-1}(\|Z_i - Z_j\|) \right)^{1-Y_{ij}} \\ \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} (Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) \right\}$$

Conditionals: Latent variable Z

$$f_{Z_i|\theta, Y}(Z_i|\theta, Y) \propto \prod_{j \neq i} \left(\text{logit}^{-1}(\|Z_i - Z_j\|) \right)^{Y_{ij}} \left(1 - \text{logit}^{-1}(\|Z_i - Z_j\|) \right)^{1-Y_{ij}} \\ \exp \left\{ -\frac{1}{2\sigma_{k_i}^2} (Z_i - \mu_{k_i})^T (Z_i - \mu_{k_i}) \right\}$$

Do not know how to sample directly from this distribution, hence MH step.
Symmetric proposal (deviating from Hoff):

$$q(Z_*|\theta^{(t)}, Y) \sim \text{MVN}_d(0, I_d) \\ R(Z^*, Z^{(t)}) = \frac{f_{Z|\theta, Y}(Z^*|\theta^{(t)}, Y)q(Z^{(t)}|\theta^{(t)}, Y)}{f_{Z|\theta, Y}(Z^{(t)}|\theta^{(t)}, Y)q(Z^*|\theta^{(t)}, Y)} \\ = \frac{f_{Z|\theta, Y}(Z^*|\theta^{(t)}, Y)}{f_{Z|\theta, Y}(Z^{(t)}|\theta^{(t)}, Y)}$$

MCMC Pseudocode

Algorithm 3: Gibbs sampler for latent network model

1 **LNM MCMC** (G, N_k, d, ns);

Input : Graph G

Number of groups N_k

Dimension of Latent Variable d

Number of samples ns

Output: Posterior $p(Z|Y, \theta)$

2 Initialize parameters $\mu^{(0)}, \sigma^{2(0)}, \lambda^{(0)}, K^{(0)}, Z^{(0)}$;

3 **for** $t = 2, \dots, ns$ **do**

4 **for** $k = 1, \dots, N_k$ **do**

5 sample $\mu_k | \theta^{(t)}, Y \sim$

$$MVN_d \left(\sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\} \frac{Z_i^{(t-1)}}{(\sigma_{k_i}^2)^{(t-1)+1}}, \sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\} \frac{(\sigma_{k_i}^2)^{(t-1)}}{(\sigma_{k_i}^2)^{(t-1)+1}} I_d \right);$$

6 **end**

7 **for** $k = 1, \dots, N_k$ **do**

8 sample $\sigma_k^2 | \theta^{(t)}, Y \sim \left(1 + \sum_{i=1}^{N_v} \mathbb{I}\{k_i = \right.$

$$k\} (Z_i^{(t-1)} - \mu_k^{(t)})^T (Z_i^{(t-1)} - \mu_k^{(t)}) \Big) \text{Inv}\chi^2_{1+d \sum_{i=1}^{N_v} \mathbb{I}\{k_i = k\}};$$

9 **end**

10 **for** $i = 1, \dots, N_v$ **do**

11 sample $K_i \sim \text{Multinoulli}(G, \lambda^{(t)})$;

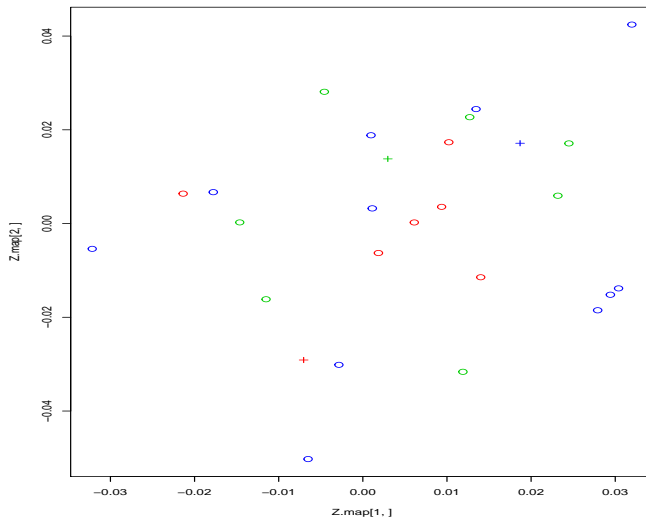
12 **end**

```

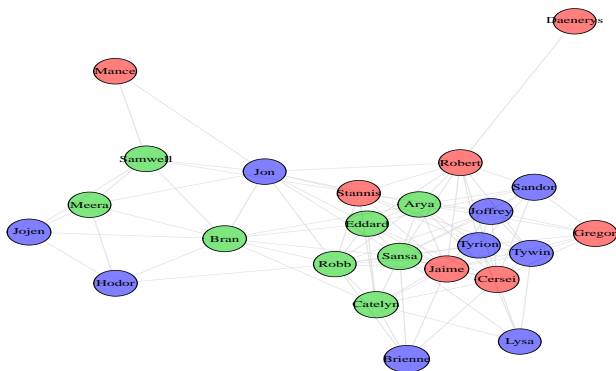
10  for  $i = 1, \dots, N_v$  do
11      | sample  $K_i \sim \text{Multinoulli}(G, \lambda^{(t)});$ 
12  end
13  for  $i = 1, \dots, N_v$  do
14      | sample  $Z_i^* \sim \text{MVN}_d(0, I_d);$ 
15      |  $R(Z_i^*, Z_i^{(t)}) = \min \left( 1, \frac{f_{Z|\theta, Y}(Z_i^* | \theta^{(t)}, Y, Z_{[-1]})}{f_{Z|\theta, Y}(Z_i^{(t)} | \theta^{(t)}, Z_{[-1]})} \right);$ 
16      | sample  $U \sim \mathcal{U}(0, 1);$ 
17      | if  $U \leq R(Z_i^*, Z_i^{(t)})$  then
18          |  $Z_i^{(t+1)} = Z_i^*;$ 
19      | else
20          |  $Z_i^{*(t+1)} = Z_i^{(t)}$ 
21      | end
22  end
23  end
24 end

```

MCMC: Mean MAP Estimates



MCMC: Clustering







EM VS MCMC

- ① Results from EM using weighted network yields more information than the unweighted
- ② Both EM algorithms converge much faster than the MCMC for this model
- ③ Large room for the MCMC to improve:
 - i Block-update covariate coefficients with the scale of latent space positions
 - ii This model easily extends to the regression framework were we can perform further analysis with the incorporation of vertex attributes

Conclusion

- 1 The information we can draw from both EM and MCMC is very interpretable. We can group characters by their geographical location, or personal relation; we can also detect if a character stands out (i.e. Daenerys).
- 2 This project used many topics covered in class; Newton-Raphson, EM, MCMC, Graphical models, Ise, and sampling

References

-  Andrew Beveridge and Jie Shan. “Network of thrones”. In: *Math Horizons* 23.4 (2016), pp. 18–22.
-  Peter D Hoff, Adrian E Raftery, and Mark S Handcock. “Latent Space Approaches to Social Network Analysis”. In: *Journal of the American Statistical Association* 97.460 (2002), pp. 1090–1098. DOI: 10.1198/016214502388618906.
-  Pavel N. Krivitsky et al. “Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models”. In: *Social Networks* 31.3 (2009), pp. 204–213. ISSN: 0378-8733.
-  Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. “On Spectral Clustering: Analysis and an Algorithm”. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. NIPS'01. MIT Press, 2001, pp. 849–856.