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

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 $\quad N_V(G) = 107$

ii Interactions $\rightarrow$ edges $\quad N_E(G) = 352$

iii numbers of interaction $\rightarrow$ Weights on the edge

Form a Network
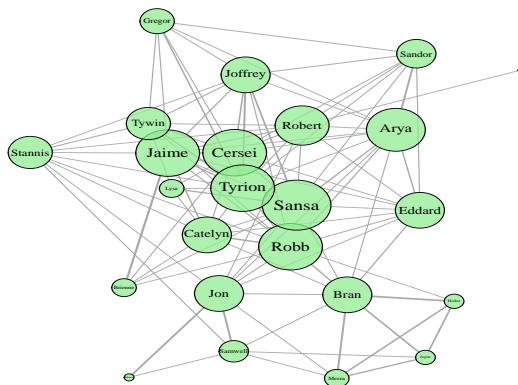
a Q: How to handle the sparsity of the network?

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 ith and jth character

   ii i.e., $a(i,j) = 0$ if there's no interaction between the two people

Unfiltered Network
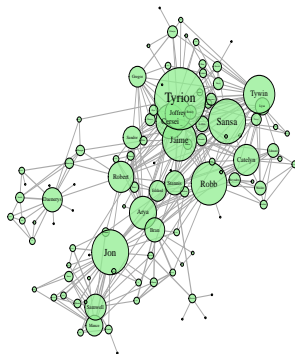
Filtered Network

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

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

Latent Network Model: Priors

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

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

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

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

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

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

Latent Network Model: Likelihood

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

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 **Y** and parameters $\theta = (\alpha, \beta)$

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

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:

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

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

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   <u>LNM EM</u> (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 $\pi^{(t)}$, $\eta^{(t)}$;
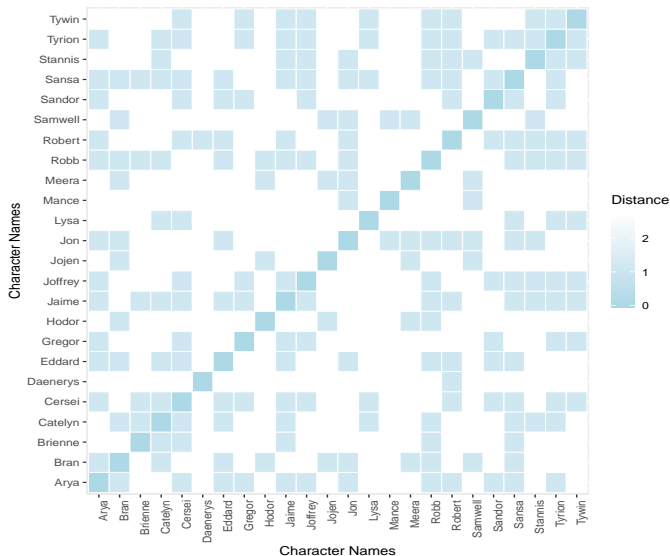
4   **M:** update $\alpha^{(t+1)}$ using (α_U);

5   update $\beta^{(t+1)}$ using (β_U);
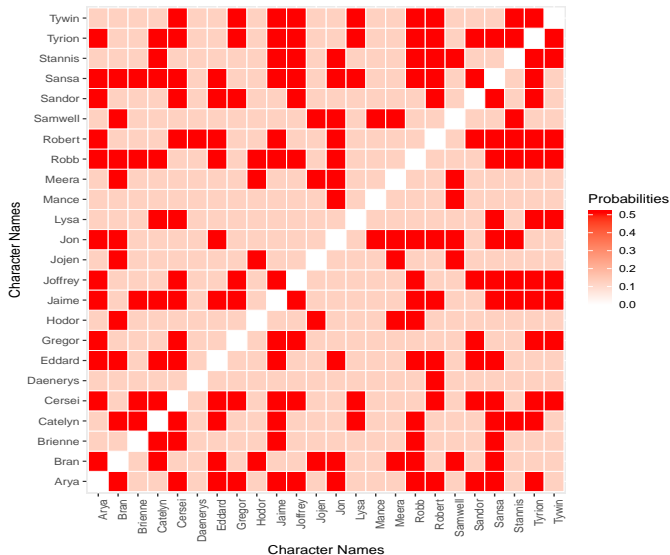
6   calculate $Q(\theta, \theta^{(t+1)})$

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

8   **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: Distances

Unweighted Network Model: Probabilities

Weighted Network Model

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

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

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

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

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

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: Psuedocode

Algorithm 2: EM for simplified latent network weighted model

---

1 <u>LNM EM</u> (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);
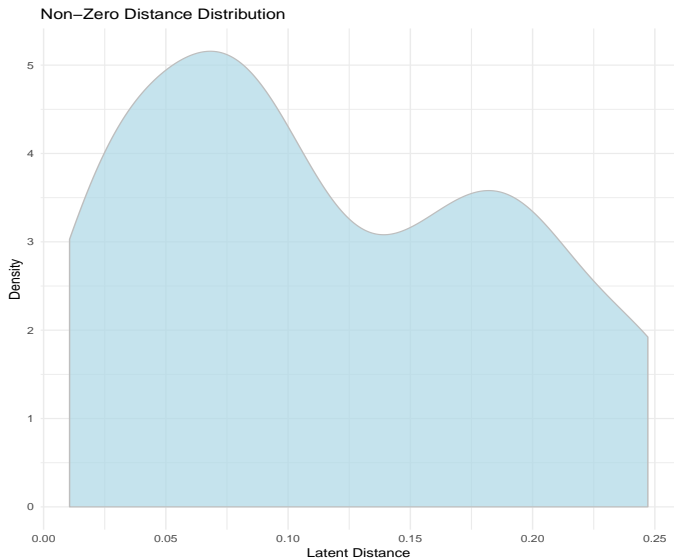
6   calculate $Q(\theta, \theta^{(t+1)})$

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

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

---

Weighted Network Model: Distance Estimates

Weighted Network Model: Distance Density

Non−Zero Distance Distribution

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

3. Interpret these estimates as smooth estimates of the weighted - Adjacency matrix

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

4. Spectral Clustering Visualization

Weighted Network Model: Spectral Clustering
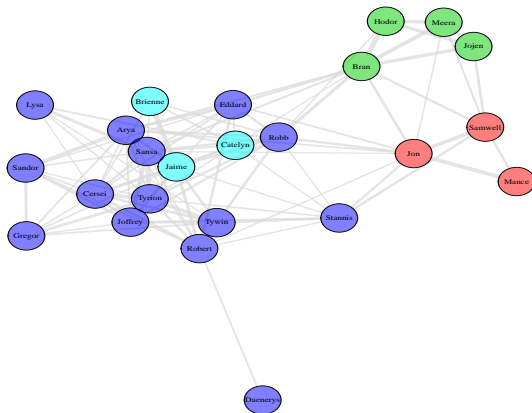
Latent Network Model

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

$$Z_i|K_i = k_i \overset{ind}{\sim} MVN(\mu_{k_i}, \sigma^2_{k_i} I_d)$$

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

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

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

$$\sigma^2_k \overset{iid}{\sim} \text{Inv}\chi^2_1$$

Conditionals: μ

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

$$\propto \exp\left\{\sum_{i=1}^{N_v}\mathbb{I}\{k_i = k\}\left[-\frac{(\sigma_{k_i}^2 + 1)}{2\sigma_{k_i}^t}\left(\mu_{k_i} - \frac{Z_i}{(\sigma_{k_i}^2 + 1)}\right)^T\left(\mu_{k_i} - \frac{Z_i}{(\sigma_{k_i}^2 + 1)}\right)\right.\right.$$

Thus for all $k \in \{1, \ldots, G\}$, 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

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

$$\propto (\sigma^2_{k_i})^{\left((\frac{-c-1}{2}ng-ng+1)-1\right)}\exp\left\{-\frac{1}{2\sigma^2_{k_i}}\sum_{K_i=k}\left((Z_i - \mu_{k_i})^T(Z_i - \mu_{k_i})+1\right)\right\}$$

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

$$\sigma^2_k|\theta^{(t)}, Y \sim \text{Inv}\Gamma(\frac{c}{2}, \frac{1}{2}) \overset{D}{=} \tau^2 \, \nu \, \text{Inv}\chi^2_c$$

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

$$\nu_{post} = (c+1)n_g + 2(n_g - 1)$$

$$\tau^2_{post} = \frac{SS_g + n_g}{(c+1)n_g + 2(n_g - 1)},$$

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:

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

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):

$$q(Z_*|\theta^{(t)}, Y) \sim MVN_2(0, I_2)$$

$$R(Z^*, Z^{(t)}) = \frac{f_{Z|\theta,Y}(Z^*|\theta^{(t)}, Y)q(Z^{(t)}|\theta^{(t)}, Y)}{f_{Z|\theta,Y}(Z^{(t)}|\theta^{(t)}, Y)q(Z^*|\theta^{(t)}, Y)}$$

$$= \frac{f_{Z|\theta,Y}(Z^*|\theta^{(t)}, Y)}{f_{Z|\theta,Y}(Z^{(t)}|\theta^{(t)}, Y)}$$

MCMC Pseudocode

Algorithm 3: Gibbs sampler for latent network model

1   <u>LNM MCMC</u> (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** $t = 2, \ldots, ns$ **do**

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

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

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

9   **end**

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

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

12   **end**

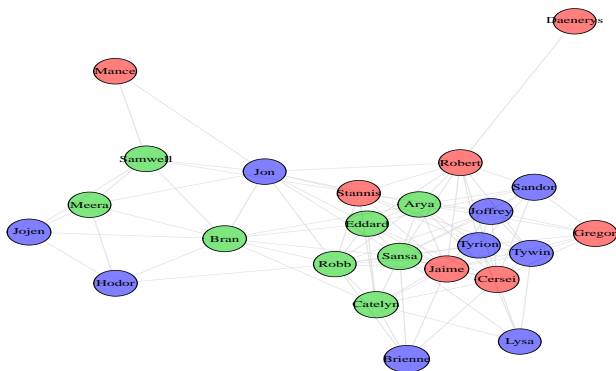```
10   for i = 1, . . . , N_v do
11   sample K_i ∼ Multinoulli(G, λ^(t));
12   end
13   for i = 1, . . . , N_v do
14   sample Z_i^* ∼ MVN_d(0, I_d);
15   R(Z_i^*, Z_i^(t)) = min (1, f_{Z|θ,Y}(Z_i^*|θ^(t),Y,Z_[-1]) / f_{Z|θ,Y}(Z_i^(t)|θ^(t),Z_[-1])) ;
16   sample U ∼ U(0, 1);
17   if U ≤ R(Z_i^*, Z_i^(t)) 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
```

$$R(Z_i^*, Z_i^{(t)}) = \min \left(1, \frac{f_{Z|\theta,Y}(Z_i^*|\theta^{(t)},Y,Z_{[-1]})}{f_{Z|\theta,Y}(Z_i^{(t)}|\theta^{(t)},Z_{[-1]})}\right)$$

# MCMC: Estimated Means
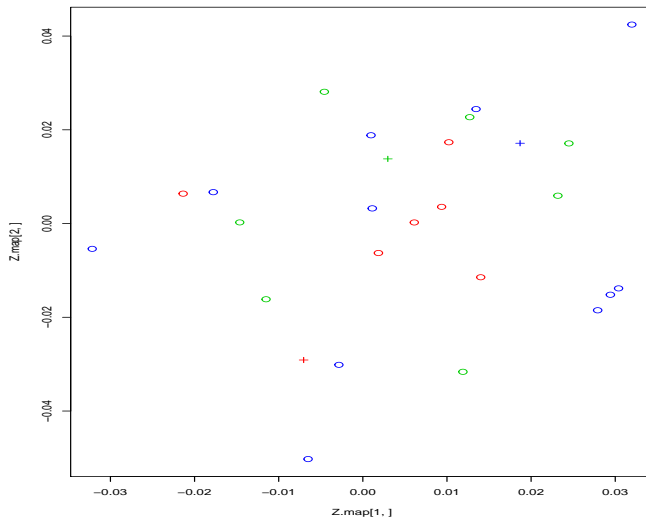
# MCMC: Clustering

# EM VS MCMC

a results from EM using weighted network has more information than the unweighted

b both EM algorithms converge much faster than the MCMC for this model

c large room for the MCMC to improve :

   i block-update covariate coefficients with the scale of latent space positions

   ii once the distributions of the variables are known, we can perform further analysis such as regression

# 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 had used many subjects covered in class Newton- Raphson, EM, MCMC, Graphical model, lse, and sampling – the material from class is very useful!