Modelling and analysis of social network data from rank preferences

Pierre OSSELIN

St Peter's College

University of Oxford

A thesis submitted for the degree of

MSc in Mathematical Sciences

Trinity 2018

Contents

Chapter 1

Introduction

This paper aims at performing inference on social network through partial ranking of the individuals

We use the Plackett-Luce model:

$$p(\rho_i|(\lambda)) = \prod_{k=1}^{K} \frac{\lambda_{i\rho_k}}{\sum_{j\neq i} \lambda_{i,j} - \sum_{l=1}^{k-1} \lambda_{i,\rho_l}} \tag{1.1}$$

Where K is the number of ranked individuals. This model is equivalent to successively choose our favorite individual by pondering our strength of affiliation with each person, and take randomly the best remaining by taking into account the proportion of this strength among the other, which is a conceivable model.

The first thing to do is to prove that Plackett-Luce is well defined as a probability distribution. It is easy to prove it by induction over $K \leq n$.
We denote $\Omega_{K,n} := \{\rho : \{1, ..., K\} \to I_K \ bijective, I_K \in [n]^{(K)}\}$. For $K = 1$:

$$\sum_{\rho\in\Omega_{1,n}} p(\rho|(\lambda)) = \sum_{k\neq i} \frac{\lambda_{i,k}}{\sum_{j\neq i} \lambda_{i,j}} = 1 \tag{1.2}$$

If the result is true for $K - 1 < n$ then, with
$\forall j \neq i, \Omega_{K,n,j} = \{\rho : \{1, ..., K\} \to I_K \ bijective, I_K \in [n]^{(K)}, \rho(1) = j\}$ and $(\hat{\lambda})_j$ the matrix without column j:

$$\sum_{\rho\in\Omega_{K,n}} p(\rho|(\lambda)) = \sum_{j=1}^{n} \sum_{\rho\in\Omega_{K,n,j}} p(\rho|(\lambda)) \tag{1.3}$$

$$= \sum_{j=1}^{n} \frac{\lambda_{i,j}}{\sum_{l\neq i} \lambda_{i,l}} \underbrace{\sum_{\rho\in\Omega_{K-1,n-1}} p(\rho|(\hat{\lambda})_j)}_{=1 \text{ by induction}} = 1 \tag{1.4}$$

Chapter 2

Community representation

2.1 Mathematic Formulation

Here we suppose that the strength between individuals are characterized by their similitude to certain communities.

We introduce p the number of communities and $(w_{i,k})_{1\leq i\leq n, 1\leq k\leq p}$ the strength of affiliation of the individual i to the community p.

$$\lambda_{i,j} = \sum_{k=1}^{p} w_{i,k} w_{j,k} \tag{2.1}$$

This formulation is equivalent to the rank matrix approximation often encountered in the literature where we assume that:

$$(\lambda) = WW^T \tag{2.2}$$

where W is the matrix of the $(w_{i,k})$

We suppose here that the strength of the relations are only determined by the proximity with a certain community hence that the relations are reciprocal. From the reordering inequality, we can observe that $\lambda_{i,j}$ is maximized when both i and j have the same community ranking preferences. We can interpret the formula $\lambda_{i,j} = \sum_{k=1}^{p} w_{i,k} w_{j,k}$ also as a scalar product in the p-dimensional space of communities. In this space, each individual is a point represented by a vector $w_i \in (\mathbb{R}^+)^p$ the strength of affiliation is also computed by $\lambda_{i,j} = ||w_i||.||w_j||.cos(\theta)$ we can interpret the norm of the vector as the individual global influence or sociability, and the θ as the divergence in personality. Hence a given individual will appreciate another individual with great sociability but different personality or views as much as a low sociability individual with the same mindset. In this interpretation, the prior function attributed to the $w_{i,k}$ is equivalent to putting a topography on the space, allowing points to lie more likely in a certain area. This is illustrated in the Figure 2.1

The main issue regarding this geometric approach is that having a bigger affiliation means that the community points have a greater scalar product, but doesn't mean those points are nearer. If we wanted to use a clustering approach to this problem,

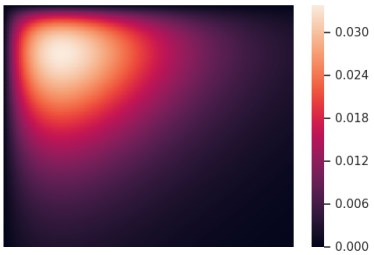

Figure 2.1: Topography of the community space for the same gamma distribution

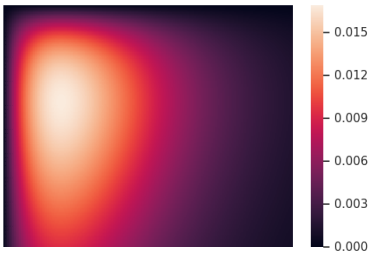

Figure 2.2: Topography of the community space for different gamma distribution

we should build a space in which the mapping to this space make it equivalent having great scalar product in the input space and being near in the output space.

2.2 Computation of the posterior

Let's define the log-posterior of the community representation. Which is the sum of the log priors of the parameters and the log-likelihood.

$$l((w);D) = \sum_{i=1}^{n}\sum_{k=1}^{p}(\alpha-1)ln(w_{i,k}) - \beta w_{i,k} + \sum_{i=1}^{n}\sum_{j=1}^{K} ln(\lambda_{i,\rho_j}) - ln(\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l}) \tag{2.3}$$

To make the optimization easier, we can compute the gradient of the posterior according to the parameter $w_{r,s}$

With

$$\frac{\partial \lambda_{i,j}}{\partial w_{r,s}} = 1_{i=r}w_{j,s} + 1_{j=r}w_{i,s} \tag{2.4}$$

We have :

$$\begin{aligned}\frac{\partial l((w);D)}{\partial w_{r,s}} = \frac{a-1}{w_{r,s}} - b + \sum_{j=1}^{K}\frac{1}{\lambda_{r,\rho_j^{(r)}}}w_{\rho_j^{(r)},s} + \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\lambda_{i,r}}w_{i,s}1_{\rho_j^{(r)}=r} - \\ \sum_{j=1}^{K}\frac{1}{\sum_{j\neq r}\lambda_{r,j} - \sum_{l=1}^{j-1}\lambda_{r,\rho_l^{(r)}}}\sum_{j\neq r}w_{j,s} - \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l^{(i)}}}w_{i,s} + \\ \sum_{j=1}^{K}\frac{1}{\sum_{j\neq r}\lambda_{r,j} - \sum_{l=1}^{j-1}\lambda_{r,\rho_l^{(r)}}}\sum_{l=1}^{j-1}w_{\rho_l^{(r)},s} + \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l^{(i)}}}\sum_{l=1}^{j-1}1_{\rho_l^{(i)}=r}w_{i,s}\end{aligned} \tag{2.5}$$

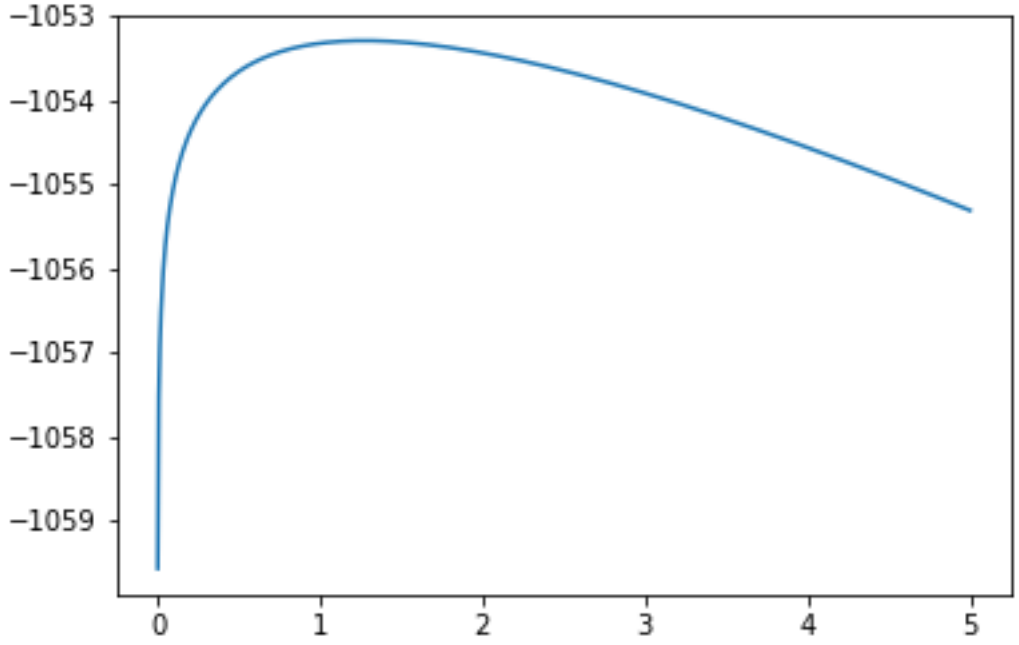

Figure 2.3: Posterior according to one coordinate

2.3 Simulations on simulated Data

2.3.1 Fixed parameters for the prior

We derived two optimization algorithm to find the MAP, one based on coordinate-wise optimization, given that for the a and b considered, the log-posterior seems coordinate-wise concave, hence assuring a more stable convergence. The other algorithm is based on the L-BFGS-B algorithm, popular in machine learning optimization.

For this simulation we took $a = 2$, $b = 1$ $n = 15$, $p = 3$, $k = 3$, drew a sample from w, computed λ and ρ. Then we maximized the posterior according to the two methods described above, in order to estimate the community strengths and then the individual mutual strength.

In figure 2.4 we can see that the coordinate and global optimization have converged to the same result. We obtain an absolute error of 0.02. Which is equivalent to 0.3 reported to the number of individual, and a ranking gap (mean absolute ranking error) of 2.13 for this individual.

It is interesting to compare the ranking gap to the expected ranking gap from a random permutation. We can manually compute with a tedious calculation that

$$\frac{1}{n}\mathbb{E}_{\sigma_n}[\sum_{k=1}^{n}|\sigma(k)-k|] = \frac{n^2-1}{3n} \tag{2.6}$$

For n = 18 we have a random expectation of the ranking gap of 6. Hence the

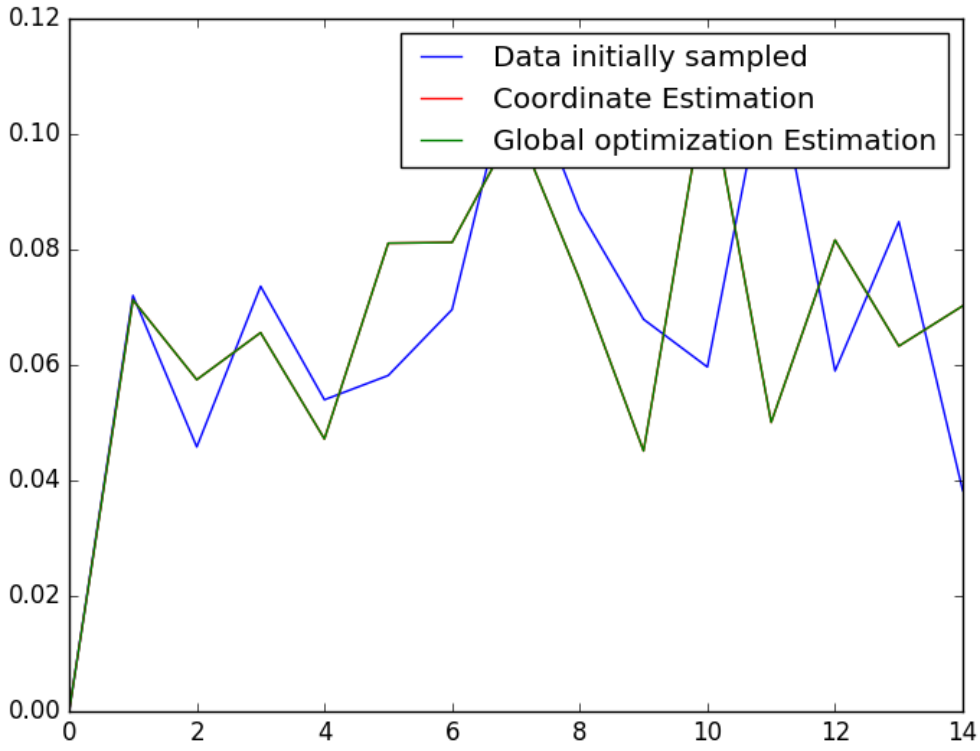

Figure 2.4: Comparison between the sampled and estimated Social affiliation of the first individual

model tends to preserve the "global" social network for each individual.

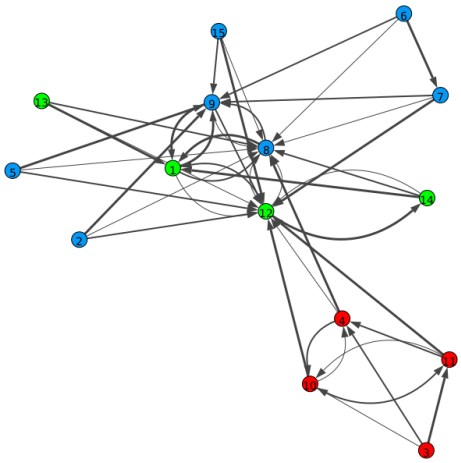

Figure 2.5: Original Community description

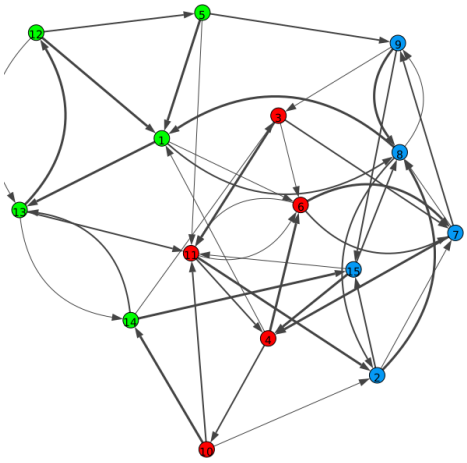

Figure 2.6: Inferred community

We can see that the inferred communities are quite similar to the original one. The original red and green communities are preserved except for the individual 6 and 5 that has passed from the blue community to the red and green respectively. Here we have assigned an individual to his preferred community, the width of an edge represents the importance the son represents for the father.

2.3.2 Model with degree-correction

In this subsection, we try to capture the different popularity that a community can have over the other communities. In the previous model, the regularization terms $w_{i,j}$ in the posterior had the same weights, irrespective of the community concerned (Having a too big $w_{i,j}$ was equally improbable regardless of the community). In a model with degree-correction, i.e in which $w_{i,k} \sim \Gamma(\alpha, \beta_k)$, we do not prevent the possibility of a popular community in the regularization, which occurs by the emergence of popular individuals.

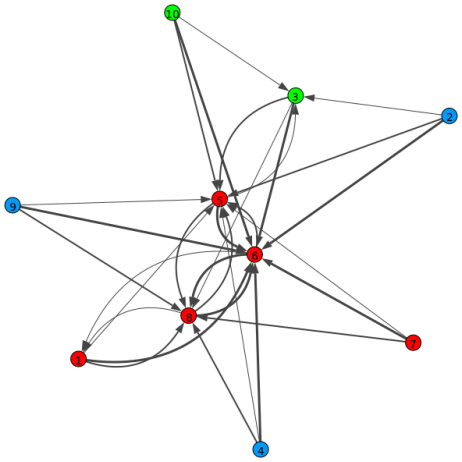

Figure 2.7: Original Community description for the degree Correction Model

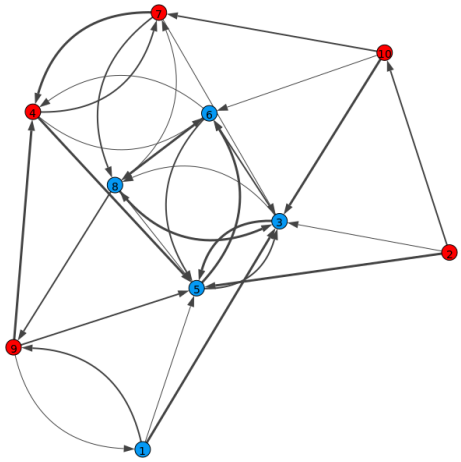

Figure 2.8: Inferred community for the degree correction Model

We can see in this example that the green community has been absorbed in an other community by the inference, and that 7 has been replaced by 3 in the other community.

2.3.3 Performance Comparison

To evaluate the performance of the models, we make prediction on the preferences for individuals who have not indicated their preferences, provide further preferences from individuals who did, or extract from the lists interpretable information about the structure of the underlying friendship network, e.g. in terms of latent communities. For this we will use different metrics. The ranking gap between latent appreciation, the absolute error in the strengths and the missed community. To do that, we will sample the strengths between individuals $\lambda_{i,j}$ and then sample full rankings from it. From this ranking we then hide a certain part of it, we feed the remaining parts to our model and then compute the error in terms of metrics previously defined to evaluate the performance.

Chapter 3

Model with Covariates

The purpose of ERGMs, in a nutshell, is to describe parsimoniously the local selection forces that shape the global structure of a network. To this end, a network dataset may be considered like the response in a regression model, where the predictors are things like propensity for individuals of the same sex to form partnerships or propensity for individuals to form triangles of partnerships. Thus, an ERGM can help us quantify the strength of this intra-group effect. The information gleaned from use of an ERGM may then be used to understand a particular phenomenon or to simulate new random realizations of networks that retain the essential properties of the original. Handcock, Hunter, Butts, Goodreau, and Morris (2008) say more about the purpose of modeling with ERGMs; yet in this article, we focus primarily on technical details.

3.1 Notation

Let N be the set of nodes in the network of interest, indexed $\{1, ..., n\}$ The relationships in the network may be directed (e.g., friendship nominations, messages) or undirected (e.g., sexual partnerships, conversations). In the former case, we define the set of dyads (here used to refer to potential relationships) $\mathbb{Y}$ to be a subset of $N \times N$, the set of ordered pairs of nodes; in the latter case, it is a subset of $\{\{i, j\} : (i, j) \in N \times N\}$, the unordered pairs of nodes. (We will also use $\mathrm{u}(\mathbb{Y})$ to refer to an unordered version of $\mathbb{Y}$, i.e., $\mathrm{u}(\mathbb{Y}) \equiv \{\{i, j\} : (i, j) \in Y\}$.) Usually, $\mathbb{Y}$ is further constrained in that in most social networks studied, a node cannot have a relationship of interest with itself, excluding pairs of the form (i, i). For binary networks, in which the relationship of interest must be either present or absent, we use $\mathcal{Y} \in 2^{\mathbb{Y}}$, the set of subsets of $\mathbb{Y}$, to refer to the set of possible networks of interest, which may be further constrained (that is, $\mathbb{Y}$ may be a proper subset of $2^{\mathbb{Y}}$). We will use $\mathbf{Y}$ to refer to network random variables and $y \in \mathcal{Y}$ to refer to their realizations, and $y_{i,j}$ be an 0 - 1 indicator of whether a relationship of interest is present between i and j in a binary network context.

3.2 Classic Graphical Models

3.2.1 ERGM model

In the ERGM model (exponential-family random graph models), the distribution of $\mathbf{Y}$ can be parameterized in the form:

$$P_\theta(Y = y) = \frac{exp((\eta(\theta))^T * z(x))}{\chi(\theta)}, y \in \mathcal{Y} \tag{3.1}$$

where θ is a q-vector of model parameters, which are mapped to a p-vector of natural parameters by $\eta(.)$, and $g(.)$ is a p-vector of sufficient statistics, which capture network features of interest. These sufficient statistics typically embody the features of the network of interest that are believed to be significant to the social process which had produced it, such as degree distribution (e.g., propensity towards monogamy in sexual partnership networks), homophily (i.e., birds of a feather flock together), and triad-closure bias (i.e., a friend of a friend is a friend) . (Morris, Handcock and Hunter, 2008)

3.2.2 The $p1$ Model

The p_1 model is the first to propose log-linear models for social networks. Suppose that we take $\mathcal{Y}$ to be the set of all directed graphs, with independent dyads [i.e., the pairs ($Y_{i,j}$, $Y_{j,i}$) are independent for different choices of $\{i, j\}$] and the following model for tie probabilities:

$$P(Y_{i,j} = y_1, Y_{j,i} = y_2) = exp\{y_1(\mu + \alpha_i + \beta_j) + y_2(\mu + \alpha_j + \beta_i) + y_1 y_2 \rho\} \backslash k_{ij}$$
$$y_1, y_2 = 0, 1; i, j = 1, ..., n; i \neq j \tag{3.2}$$

In this model, we can interpret μ as a density parameter, constituting an overall mean, α_i as a productivity parameter, characterizing i as a sender, β_j as an attractiveness parameter, characterizing j as a receiver, and ρ as a parameter indicating the force of reciprocation.

3.2.3 The $p2$ Model

While ERGMs are useful for modeling global network characteristics, models based on conditional independence (given latent variables) are useful for multiple reasons. First, ERGMs are not well understood and sometimes possess undesirable properties, e.g., model degeneracy. Second, the likelihood function of ERGMs is intractable, complicating statistical computing. Third, there may be unobserved heterogeneity or unobserved structure. Models based on independence conditional on latent variables: random effects models and mixed effects models and extensions; stochastic block models and extensions, including mixed membership models; and latent space models and extensions.

The aim of the p_2 model is to relate binary network data to covariates while taking into account the specific network structure. This requires a kind of bivariate logistic regression model capable of handling the dependence of network data. The p_2 model is based on the same formula. The density and reciprocity parameters, however, are allowed to vary over dyads, and are therefore denoted μ_{ij} and ρ_{ij}. Moreover, parameters α_i , β_j , μ_{ij} , and ρ_{ij} are further modelled using covariates. We first consider the node-specific parameters α_i and β_j. Covariates and random effects are included in a linear regression model for the productivity and attractiveness parameters.

$$\begin{cases} \alpha = X_1\gamma_1 + A \\ \beta = X_2\gamma_2 + B \end{cases} \tag{3.3}$$

This formulation expresses the plausible idea that attractiveness (or popularity) and productivity (or sociability) depend on actor attributes (denoted by X_1 and X_2 , respectively, where the same or different attributes may be used for attractiveness and productivity) with corresponding weights γ_1 and γ_1 . Naturally, the attributes do not explain all variation in attractiveness and productivity parameters, as is represented by the residual terms A and B, n $\times$ 1 vectors with components A_i and B_i . The residuals are modeled as normally distributed random variables with expectation 0 and variances σ_A^2 and σ_B^2 , respectively. Parameters σ_A^2 and σ_B^2 can be interpreted as unexplained variance, that is, the variance of the α's and β's that is left after taking into account the effect of the covariates X_1 and X_2 . The productivity and attractiveness parameters of the same node are correlated:$\forall i, cov(A_i, B_i) = \sigma_{AB}$. Independence is assumed for parameters of different actors by setting $cov(A_i, A_j) = cov(B_i, B_j) = cov(A_i, B_j) = 0$ for $i \neq j$ (cf. Wong, 1987). If no external information on actors is available, the terms $X_1\gamma_1$ and $X_2\gamma_2$ vanish and σ_A^2 and σ_B^2 denote the variances of the α and β parameters, respectively. Then a pure random effects model results with, apart from the density and reciprocity parameters, only two variance parameters and one covariance parameter. Thus, the p_2 model without covariates is a more parsimonious model than the p_1 model with well-interpretable parameters. Obviously, the fit of p_1 will be better than the fit of p_2 without covariates.

3.2.4 The SRM Model

ERGM models models work very poorly, the learned parameters do not generate data that resembles the input and tend toward wholly connected or completely empty graphs. Similarly, Since a binary random graph model does not accommodate rank data, analysis of these data with such a model typically begins by reducing all positive ranked relations to ones, and all negative ranked relations to zeros, leaving unranked relations as zeros as well. Such a data analysis essentially throws away some of the information in the data. Furthermore, the support of most binary random graph models consists of all possible graphs on the node set. In contrast, the data collection scheme used by Sampson (1969) was censored, as no graph with more than three outgoing edges per node could have been observed.

We take the same formulation as for the community representation. Latent variable $\mathbf{Y}$ representing the strength of affiliation between individuals. And $\mathbf{S}$ the sociomatrix given in data. $\mathbf{S} = \{s_{i,j} : i \neq j\}$, coded so that $s_{i,j} = 0$ if j is not nominated by i, $s_{i,j} = 1$ if j is i^{th} least favored nomination, and so on. Under this coding, $s_{i,j}$ ¿ $s_{i,k}$ if i scores j more highly than k, or if i nominates j but not k. Letting $a_i = \{1, ..., n\}\backslash\{i\}$ be the set of individuals whom person i may potentially nominate, each observed outdegree $d_i = \sum_{j \in a_i} 1(s_{i,j} > 0)$ satisfies $d_i \leq m$. We then write

$$s_{i,j} = [(m - rank_i(y_{i,j} + 1) \wedge 0)] \times 1(y_{i,j} > 0) \tag{3.4}$$

and its inverse:

$$s_{i,j} > 0 \Rightarrow y_{i,j} > 0 \tag{3.5}$$

$$s_{i,j} > s_{i,k} \Rightarrow y_{i,j} > y_{i,k} \tag{3.6}$$

$$s_{i,j} = 0 \text{ and } d_i < m \Rightarrow y_{i,j} \leq 0 \tag{3.7}$$

Given a statistical model $\{p(Y|\theta) : \theta \in \Theta\}$ for the underlying social relations Y, inference for the parameter θ can be based on a likelihood derived from the observed scores S. The likelihood is, as usual, the probability of the observed data S as a function of the parameter θ. To obtain this probability, let $F(S)$ denote the set of Y-values that are consistent with S in terms of associations (5)−(7) above. Since the entries of S are the observed scores if and only if $Y \in F(S)$, the likelihood is given by

$$L_F(\theta : \mathbf{S}) = Pr(Y \in F(S)|\theta) = \int_{F(S)} p(Y|\theta) d\mu(Y) \tag{3.8}$$

where is a measure that dominates the probability densities $\{p(Y|\theta) : \theta \in \Theta\}$.

Given current values of (θ, Y), one step of the Gibbs sampler for the FRN likelihood proceeds by updating the values as follows:

1. Simulate $\theta \sim p(Y|\theta)$
2. For each $i \neq j$, simulate $y_{i,j} \sim p(y_{i,j}|\theta, Y_{-(i,j)}, Y \in F(S))$ as follows:
   (a) if $s_{i,j} > 0$, simulate $y_{i,j} \sim p(y_{i,j}|Y_{-(i,j)}, \theta) \times 1(max\{y_{i,k} : s_{i,k} < s_{i,j}\} \leq y_{i,j} \leq min\{y_{i,k} : s_{i,k} > s_{i,j}\})$;
   (b) if $s_{i,j} = 0$, and $d_i < m$ simulate $y_{i,j} \sim p(y_{i,j}|Y_{-(i,j)}, \theta) \times 1(y_{i,j} \leq 0)$;
   (c) if $s_{i,j} = 0$, and $d_i = m$ simulate $y_{i,j} \sim p(y_{i,j}|Y_{-(i,j)}, \theta) \times 1(y_{i,j} \leq min\{y_{i,k} : s_{i,k} > 0\})$;

In the above steps, "$y \sim f(y)$" means " simulate y from a distribution with density proportional to $f(y)$." For each ordered pair (i, j), step 2 of this algorithm will generate a value of $y_{i,j}$ from its full conditional distribution, constrained so that conditions (5)(7) that define the FRN likelihood are met.

To deal with the previous remarks, we apply a model in the case where θ represents the parameters in the following standard regression model for relational data:

$$y_{i,j} = \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{x_{i,j}} + a_i + b_j + \epsilon_{i,j} \tag{3.9}$$

$$\begin{pmatrix} a_i \\ b_i \end{pmatrix}, i = 1, ..., n \sim \text{i.i.d normal}(0, \Sigma_{ab}) \tag{3.10}$$

$$\begin{pmatrix} \epsilon_{i,j} \\ \epsilon_{j,i} \end{pmatrix}, i = 1, ..., n \sim \text{i.i.d normal}(0, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}) \tag{3.11}$$

The additive row effect a_i is often interpreted as a measure of person is "sociability" whereas the additive column effect b_i is taken as a measure of is "popularity.". The parameter ρ represents potential correlation between $y_{i,j}$ and $y_{j,i}$. In a mixed-effects version of this model, the possibility that a persons sociability a_i is correlated with their popularity b_i can be represented with a covariance matrix Σ_{ab}. The covariance among the elements of $Y = y_{i,j} : i \neq j$ induced by Σ_{ab} and is called the SRM (Warner et al. ,1979).

3.3 Plackett-Luce Model with Covariates

To adapt a model with covariates to the Bradley-Terry likelihood we will model the strength matrix as follow:

$$ln(y_{i,j}) = \boldsymbol{\beta}^{\boldsymbol{T}} \boldsymbol{x_{i,j}} + a_i + b_j + \epsilon_{i,j} \tag{3.12}$$

$$\begin{pmatrix} \epsilon_{i,j} \\ \epsilon_{j,i} \end{pmatrix}, i = 1, ..., n \sim \text{i.i.d normal}(0, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}) \tag{3.13}$$

We will treat $\boldsymbol{\beta}$, $\boldsymbol{a}$ $\boldsymbol{b}$ and Σ_ϵ as parameters.
To perform bayesian inference, we need to compute an integral over a high dimensional, and intractable set of (λ).

$$L(\theta|(\rho)) \propto p(\theta)p((\rho)|\theta) = p(\theta) \int_{(\lambda)} p((\lambda)|\theta)p((\rho)|(\lambda)) \tag{3.14}$$

Given the observed ranks (ρ) and a prior distribution $p(\theta)$ over the parameter space, the joint posterior distribution with density $p(\theta, (\lambda)|(\rho))$ can be approximated by generating a Markov chain whose stationary distribution is that of $(\theta, (\lambda))|(\rho)$. The values of θ simulated from this chain provide an approximation to the (marginal) posterior distribution of θ given by the information from (ρ).
One such MCMC algorithm is the Metropolis Hasting Algorithm:
Given current values of (θ, Y), one step of the Metropolis - Hasting for the Plackett-Luce likelihood proceeds by updating the values as follows:

1. Define the target distribution $\pi(.)$. As we sample from the joint distribution, we can take the following target distribution.

$$p(\theta, (\lambda)|(\rho)) \propto p(\theta) \times p((\lambda)|\theta) \times p((\rho)|(\lambda)) = \pi(\theta, (\lambda)) \tag{3.15}$$

with:

(a) $p(\theta)$ prior over the parameters. We will take the following priors:

i. $\beta_i, a_i, b_i \sim \mathcal{N}(0, 10^5)$

ii. $\sigma \sim Gamma(1, 0.01)$, $\rho \sim \mathcal{U}(0, 1)$

(b)

$$\begin{pmatrix} log(\lambda_{i,j}) \\ log(\lambda_{j,i}) \end{pmatrix} |\theta, i < j \sim i.i.d\ \mathcal{N}(\begin{pmatrix} \beta^T x_{i,j} + a_i + b_j \\ \beta^T x_{j,i} + a_j + b_i \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}) \quad (3.16)$$

(c)

$$p(\rho_i|(\lambda)) = \prod_{k=1}^{p_i} \frac{\lambda_{i\rho_k}}{\sum_{j \neq i} \lambda_{i,j} - \sum_{l=1}^{k-1} \lambda_{i,\rho_l}} \quad (3.17)$$

2. Choose a proposal distribution $q(\theta^*, (\lambda)^*|\theta, (\lambda))$. It needs to be easy to simulate and determine an irreducible chain. Here we have calibrated the proposal with random simulation in order to reach an acceptance ratio of approximately 25%, as the litterature suggests. To do so we have chosen a proposal distribution for the $log(\lambda)$ and $\boldsymbol{\beta, a, b}$ of a normal distribution with standard deviation 0.125 and a uniform of amplitude 0.075 for ρ and of mean the precedent state of the Markov Chain. To reduce the possibility to be trapped in a local minima, we also added a probability of 10% that the proposal follows the prior distribution of those parameters. We calibrated all those values on random simulations to obtain an average of 25% of acceptance ratio.

3. Derive a Metropolis-Hasting Algorithm:

   (a) Sample $(\theta^*, (\lambda)^*)$ from the $q(.|\theta, (\lambda))$ previous distribution.

   (b) Sample $u \sim \mathcal{U}[0, 1]$

   (c) If $u < \alpha = min\{1, \frac{\pi(\theta^*, (\lambda)^*) q(\theta, (\lambda)|\theta^*, (\lambda)^*)}{\pi(\theta, (\lambda)) q(\theta^*, (\lambda)^*|\theta, (\lambda))}\}$ then set $(\theta, (\lambda)) = (\theta^*, (\lambda)^*)$ else keep the same value.

3.4 Simulation

3.4.1 Toy Example

We first ran the algorithm on a toy example of size $n = 10$, with a clear separation between a node very "Popular", that is to say a node appreciated by all the other node, but liking only one individual, and an other node that, on the contrary, likes everyone but nobody likes him. This node can be considered "Sociable" but not popular.

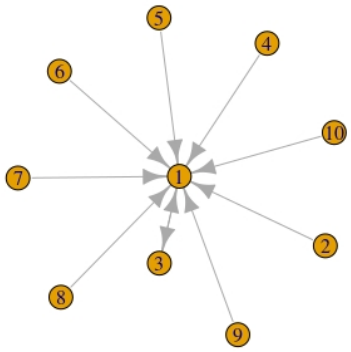

Figure 3.1: Graph Network Example

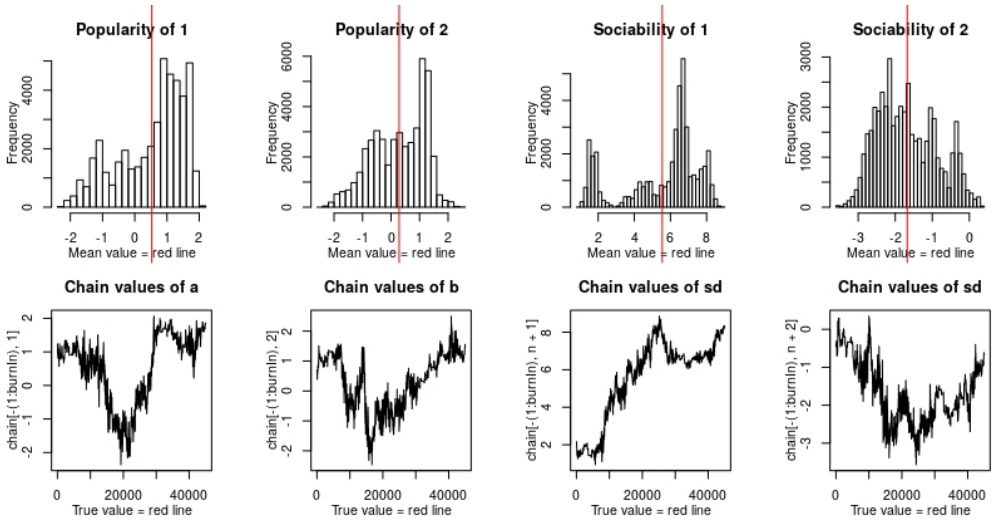

Figure 3.2: Node 1 and 2 popularity and sociability

Here we can see node 1 popularity skyrockets compared to its sociability and both node 2 sociability and popularity.

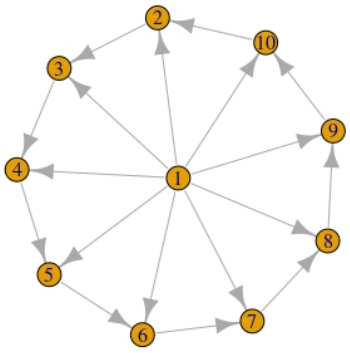

Figure 3.3: Graph Network Example

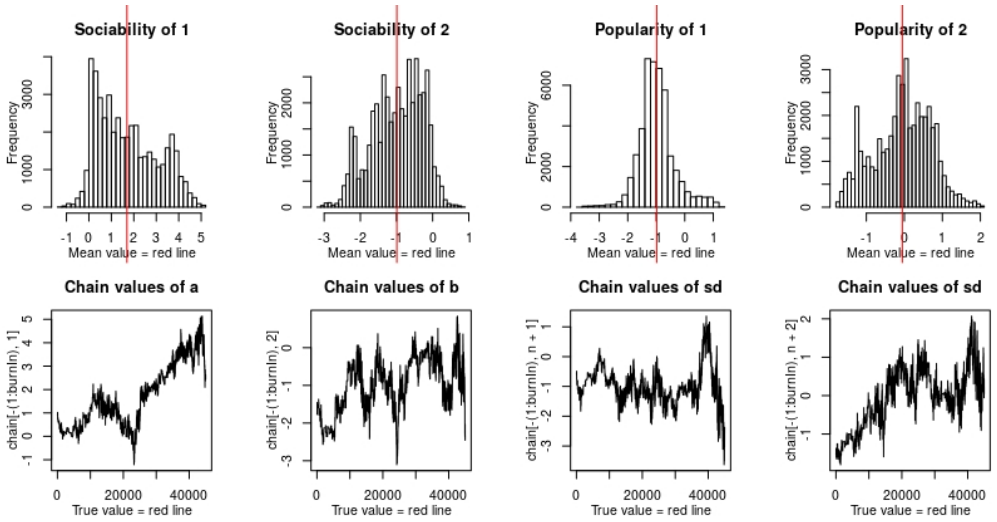

Figure 3.4: Node 1 and 2 popularity and sociability

Here we can see node 1 sociability is higher than its popularity and both node 2 sociability and popularity.

3.4.2 Sampson dataset

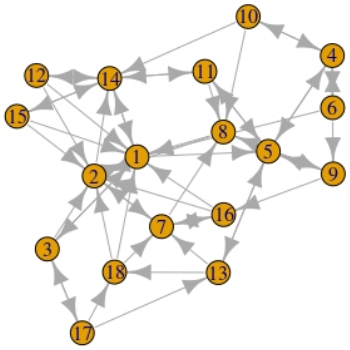

Figure 3.5: Graph Network Example

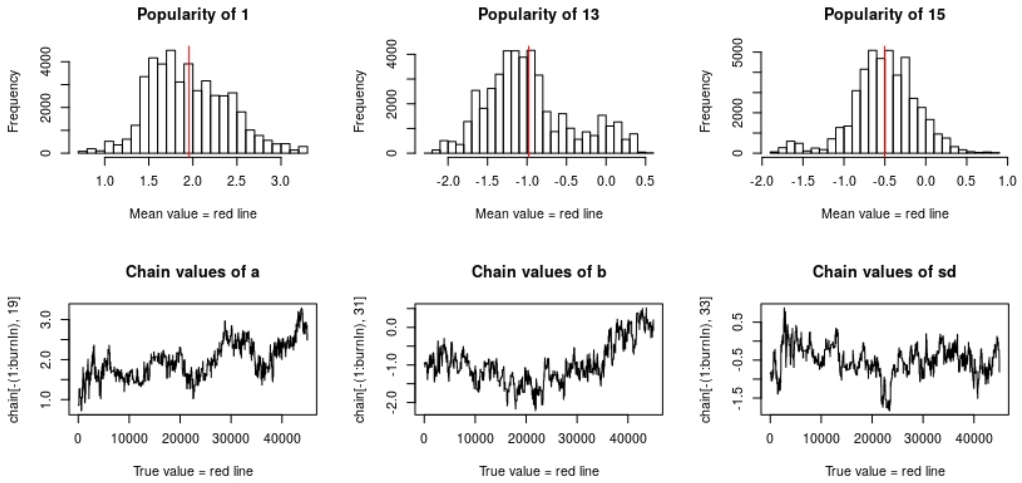

Figure 3.6: Markov Chains

Node 1 is appreciated by half the monks and has a high popularity. The two others are only appreciated by one other monks and have negative popularity.

Chapter 4

EM algorithm with community representation

We have seen that the derivation of the posterior distribution over the (w) derived in equation (2.5) is somewhat very complicated. Hence we try in this chapter to introduce latent variables in order to derive an EM algorithm.

4.1 Mathematical formulation

We recall the likelihood of the Plackett-Luce Model:

$$p(\rho_i|(\lambda)) = \prod_{k=1}^{K} \frac{\lambda_{i\rho_k}}{\sum_{j\neq i} \lambda_{i,j} - \sum_{l=1}^{k-1} \lambda_{i,\rho_l}} \tag{4.1}$$

Along with its community representation:

$$\lambda_{i,j} = \sum_{k=1}^{p} w_{i,k} w_{j,k} \tag{4.2}$$

With elements of (λ) and (w) positive.

If we add a latent variable Z with its probability distribution $P(Z|D,\lambda)$ we obtain the following complete data-likelihood:

$$L(\lambda, Z) = P(D, Z|\lambda) = (\prod_{i=1}^{n}\prod_{k=1}^{K} \frac{\lambda_{i\rho_k}}{\sum_{j\neq i} \lambda_{i,j} - \sum_{l=1}^{k-1} \lambda_{i,\rho_l}})P(Z|D,\lambda) \tag{4.3}$$

Hence our approach is to find latent variables that would both give sense and a nice posterior distribution over the (w) to derive an EM algorithm.

We first introduce:

$$V_{i,j,k} \sim \mathcal{E}(w_{i,k}w_{j,k}),\ 1 \leq i \neq j \leq n,\ \ 1 \leq k \leq p \tag{4.4}$$

$$V_{i,j} = min(V_{i,j,1}, ..., V_{i,j,p}) \sim \mathcal{E}(\lambda_{i,j}) \tag{4.5}$$

And

$$Z_{i,k}|\rho_i = min(V_{i,j})_{j\neq i,\rho_i[1],...,\rho_i[k-1]} \sim \mathcal{E}(\sum_{j\neq i,\rho_i[1],...,\rho_i[k-1]} \lambda_{i,j}), 1 \leq k \leq K \tag{4.6}$$

An interpretation of those latent variables is that the next person ranked by an individual is equivalent to the first arrived individual (among those remaining to be ranked) following a waiting time of parameter $\lambda_{i,j}$. The arriving time of this same individual is equivalent to the first arriving time of other individuals following a waiting time of parameter $w_{i,k}w_{j,k}$ which is the contribution of the k^{th} community.

This latent variable allows us to suppress the denominator in (4.1). To suppress the nominator we introduce the following latent variables:

$$Y_{i,j} \sim \mathcal{B}(\frac{w_{i,1}w_{\rho_i[j],1}}{\lambda_{i,j}}, ..., \frac{w_{i,p}w_{\rho_i[j],p}}{\lambda_{i,j}}), 1 \leq j \leq K \tag{4.7}$$

This variable can be interpreted as the explanation of the choice of a certain individual in terms of community.

We obtain the following log-likelihood distribution.

$$\mathcal{L}((w); Y, Z, D) = \sum_{i=1}^{n}\sum_{k=1}^{K}\Big\{ -(\sum_{l\neq i}\lambda_{i,l} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l})z_{i,k} + \sum_{p=1}^{P}\delta_{y_{i,k},p}(ln(w_{i,p} + ln(w_{\rho_k,p})))\Big\} \tag{4.8}$$

4.2 EM formulation

E-step:

$$\mathbb{E}_{Z,Y|\rho,(w)^*}(\mathcal{L}((w); Y, Z, D)) = \sum_{i=1}^{n}\sum_{k=1}^{K}\Big\{ -\frac{(\sum_{l\neq i}\lambda_{i,l} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l})}{(\sum_{l\neq i}\lambda^*_{i,l} - \sum_{l=1}^{k-1}\lambda^*_{i,\rho_l})} + \sum_{p=1}^{P}\frac{w^*_{i,p}w^*_{\rho_k,p}}{\lambda^*_{i\rho_k}}(ln(w_{i,p} + ln(w_{\rho_k,p})))\Big\} \tag{4.9}$$

Hence, with a gamma prior on the (w) we obtain:

$$Q(w|w^*) = \sum_{i=1}^{n}\Bigg[\sum_{k=1}^{K}\Big\{ -\frac{(\sum_{l\neq i}\lambda_{i,l} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l})}{(\sum_{l\neq i}\lambda^*_{i,l} - \sum_{l=1}^{k-1}\lambda^*_{i,\rho_l})} + \sum_{p=1}^{P}\frac{w^*_{i,p}w^*_{\rho_k,p}}{\lambda^*_{i,\rho_k}}(ln(w_{i,p} + ln(w_{\rho_k,p})))\Big\} + \sum_{p=1}^{P}(a-1)ln(w_{i,p}) - bw_{i,p}\Bigg] \tag{4.10}$$

M-step: With the same result as in chapter 2 for the full derivation:

$$\frac{\partial\lambda_{i,j}}{\partial w_{r,s}} = 1_{i=r}w_{j,s} + 1_{j=r}w_{i,s} \tag{4.11}$$

We obtain this time, after computation:

$$w_{r,s}\frac{\partial Q}{\partial w_{r,s}} = C(r,s,w^*) - A(r,s,w,w^*)w_{r,s} \tag{4.12}$$

With

$$C(r,s,w^*) = a - 1 + \sum_{k=1}^{K} \frac{w^*_{r,s}w^*_{\rho_r[k],s}}{\lambda^*_{r,\rho_r[k]}} + \sum_{i\neq r} \frac{w^*_{i,s}w^*_{r,s}}{\lambda^*_{i,r}}\mathbb{I}(r \in \rho_i) \tag{4.13}$$

and

$$\begin{aligned} A(r,s,w,w^*) = b \; + \; & \left(\sum_{k=1}^{K} \frac{1}{\Lambda^*(r,k)}\right)\left(\sum_{l\neq r} w_{l,s}\right) + \sum_{i\neq r} w_{i,s}\left(\sum_{k=1}^{K} \frac{1}{\Lambda^*(i,k)}\right) - \\ & \sum_{l=1}^{K-1} w_{\rho_r(l),s}\left(\sum_{k=l+1}^{K} \frac{1}{\Lambda^*(r,k)}\right) - \sum_{i\neq r} w_{i,s}\left(\sum_{k>rank_{\rho_i}(r)}^{K} \frac{1}{\Lambda^*(i,k)}\right)\mathbb{I}(r \in \rho_i) \end{aligned} \tag{4.14}$$

where

$$\Lambda^*(i,k) = \sum_{l\neq i} \lambda^*_{i,l} - \sum_{l=1}^{k-1} \lambda^*_{i,\rho_i[l]} \tag{4.15}$$

We can see that it is difficult to directly derive the maximum arguments for $Q(w|w^*)$, as putting the gradient to zero require to solve a set of non linear equations. However, we can notice that $A(r,s,w,w^*)$ does not depend on $w_{r,s}$. Moreover, we can easily check by remounting the equations that A and C are always positive (C might be negative if $a < 1$). Hence, as a function of one parameter, Q has one maximum reached by setting the derivative to zero. Hence instead of updating the whole set of variables w directly by maximizing Q, we can update the elements one by one and update the set of fixed elements w^*. As A and C are always positive, doing so will always increase the posterior likelihood. We can update $w_{r,s}$ with

$$w^{(t)}_{r,s} = \frac{C(r,s,w^{(t-1)})}{A(r,s,w^{(t-1)})} \tag{4.16}$$

This process will converge to (at least) a local minima.

4.3 EM formulation with variable ranks

In this section we suppose that K depends on the individual i i.e we allows the indiduals to rank as many people as they want. The resulting model is the same, with only replacing K with K_i. We obtain the following log-likelihood distribution.

$$\mathcal{L}((w); Y, Z, D) = \sum_{i=1}^{n} \sum_{k=1}^{K_i} \Big\{ -(\sum_{l \neq i} \lambda_{i,l} - \sum_{l=1}^{k-1} \lambda_{i,\rho_l}) z_{i,k} + \sum_{p=1}^{P} \delta_{y_{i,k},p} (ln(w_{i,p} + ln(w_{\rho_k,p}))) \Big\} \tag{4.17}$$

We proceed the same way as the previous section, and we obtain the same formulations by allowing different lengths for the rows of ρ. In particular, we obtain the following EM algorithm for inference on the w:

$$w_{r,s}^{(t)} = \frac{C(r, s, w^{(t-1)})}{A(r, s, w^{(t-1)})} \tag{4.18}$$

with

$$C(r, s, w^*) = a - 1 + \sum_{k=1}^{K_r} \frac{w_{r,s}^* w_{\rho_r[k],s}^*}{\lambda_{r,\rho_r[k]}^*} + \sum_{i \neq r} \frac{w_{i,s}^* w_{r,s}^*}{\lambda_{i,r}^*} \mathbb{I}(r \in \rho_i) \tag{4.19}$$

and

$$\begin{aligned} A(r, s, w, w^*) = b \; + \; & \Big(\sum_{k=1}^{K_r} \frac{1}{\Lambda^*(r,k)} \Big) \Big(\sum_{l \neq r} w_{l,s} \Big) + \sum_{i \neq r} w_{i,s} \Big(\sum_{k=1}^{K_i} \frac{1}{\Lambda^*(i,k)} \Big) - \\ & \sum_{l=1}^{K_r - 1} w_{\rho_r(l),s} \Big(\sum_{k=l+1}^{K_r} \frac{1}{\Lambda^*(r,k)} \Big) - \sum_{i \neq r} w_{i,s} \Big(\sum_{k > rank_{\rho_i}(r)}^{K_i} \frac{1}{\Lambda^*(i,k)} \Big) \mathbb{I}(r \in \rho_i) \end{aligned} \tag{4.20}$$

where

$$\Lambda^*(i,k) = \sum_{l \neq i} \lambda_{i,l}^* - \sum_{l=1}^{k-1} \lambda_{i,\rho_i[l]}^* \tag{4.21}$$

4.4 Simulation

4.4.1 Toy Example

We drew a first simple example with a clear clustering into two groups of individuals as in figure (4.1). Running the EM algorithm implemented in R is instantaneous and return the following matrix for W:

$$W = \begin{pmatrix} 0.1294021 & 1.0349799 \\ 0.1228317 & 0.8720837 \\ 0.1187870 & 0.7219490 \\ 1.0349782 & 0.1294015 \\ 0.8720818 & 0.1228311 \\ 0.7219470 & 0.1187863 \end{pmatrix} \tag{4.22}$$

which show a clear differentiation between two groups as illustrated in figure (4.2). Furthermore, we can see that the greatest values for $w_{i,j}$ in the respective communities are assigned to the most appreciated individuals in each group (denoted by bold arrows), which is consistent with what we would expect.

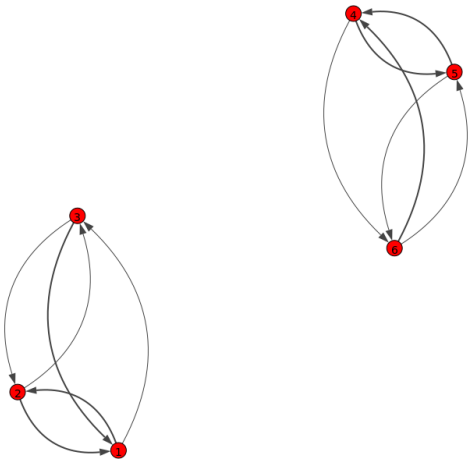

Figure 4.1: Initial Network with two independent group of individuals

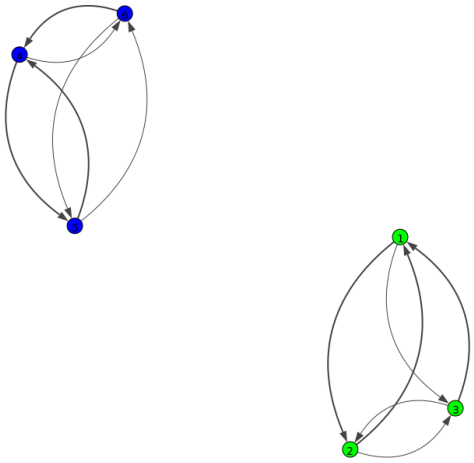

Figure 4.2: Inferred communities

4.4.2 Performance Evaluation

In this subsection we tested our EM algorithm on simulated random networks of size $n = 20, K = 5, p = 4, a = 2, b = 2$ and plotted the log posterior for each update in the matrix (w), one epoch being one full update of the matrix. We also performed the EM algorithm on the same network with different initial values of (w).

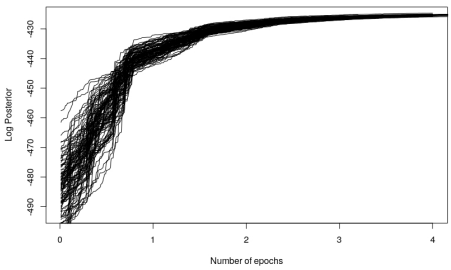

Figure 4.3: Performance on the first 4 epochs

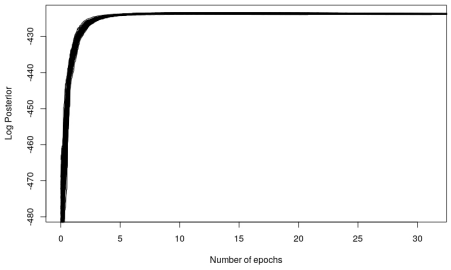

Figure 4.4: Performance on 30 epochs

We can see that the algorithm performs quite well, with three updates for each parameters $w_{i,j}$ we have a very good estimation of the maximum a posteriori.

4.4.3 Sampson Dataset

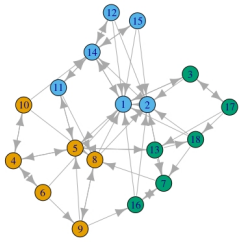

Figure 4.5: sampson clustering t = 1

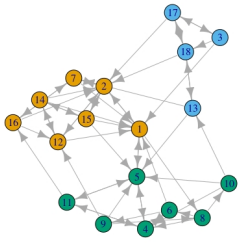

Figure 4.6: sampson clustering t = 2

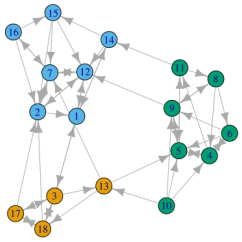

Figure 4.7: sampson clustering t = 3

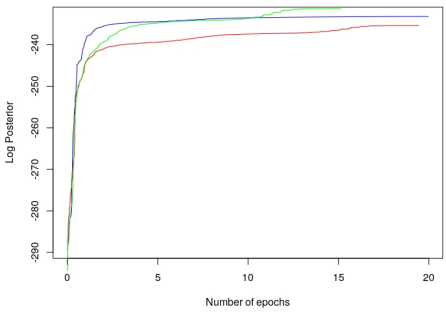

Figure 4.8: Posterior Sampson Simulation

Chapter 5

Model Test

5.1 Test on simulated Data

5.1.1 Comparison EM algorithm and optimisation algorithm

In this subsection we randomly drew graphs of size $n = 20$ and performed the EM algorithm as well as the L-BFGS-B optimization algorithm in order to compare their performances, we took the same random initialization each time. We computed the resulting posterior likelihood of the W resulting from the algorithms. And we plotted them in the same graph.

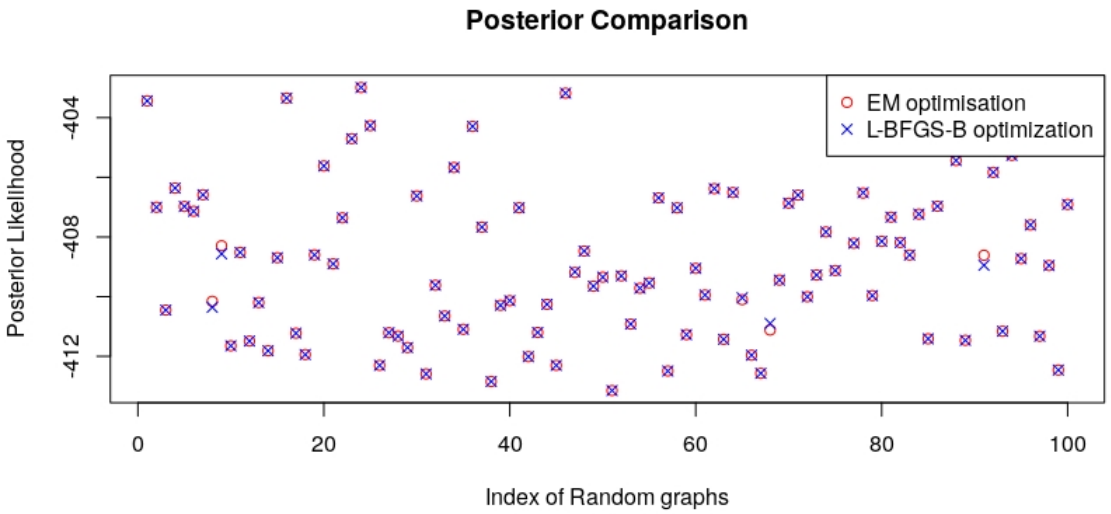

Figure 5.1: Performance Comparison between the EM and L-BFGS-B algorithm

.

5.1.2 Comparison EM algorithm and optimisation algorithm and Covariates Method

In this subsection we will compare the recovering of the lambda parameters