Modelling and analysis of social network data from rank preferences

Pierre Osselin

University of Oxford

MSc in Mathematical Sciences

Trinity 2019

Abstract

Data under the form of rank nomination networks intervene in many applications, from sociology and political studies to marketing or the examination of animal behaviour among others. The Plackett-Luce model, an extension of the Bradley Terry model for pair-wise comparisons, has been widely used to describe them. In this paper, we adapt the model to perform community detection and preference prediction through the introduction of proper parameters and latent variables which allow us to derive an EM (Expectation-Maximization) algorithm and a Gibbs Sampler to find MAP (Maximum a posteriori) estimates. We also introduce different versions of the model to deal with covariate variables, together with their bayesian inference schemes. We assess experimentally the efficiency of these algorithms on synthetic data and a variety of applications.

Contents

Chapter 1

Introduction

Given a social network of n individuals, where each person gives a list of his top K-friend in the population, let $\lambda_{i,j} > 0$ be the esteem that individual i expresses for individual j, with $i \neq j$. These quantities are not necessarily symnetrical. We consider the following generative model for the lists of preferences, called the Plackett-Luce model, and proposed by R. Duncan Luce [1] and R. L. Plackett [4]:

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

Where $\rho_i = (\rho_{i1}, ..., \rho_{ik})$ denotes the top-K list of individual i. Intuitively, this model corresponds to the situation where an individual iteratively chooses the best individual among the remaining ones as follows: Each remaining individuals j has a value $\lambda_{i,j}$ and has a probability of being chosen equal to $\lambda_{i,j} / \sum\limits_{k \in R} \lambda_{i,k}$ where R is the set of remaining individuals that have not been ranked.

This model is quite popular to learn how to rank, and has found numerous applications, including the study of political polls [22], competitive racing [17] or market behaviour [26].

Its popularity arises from its statistical foundation, and its analytic form. In particular, Luce showed that model 1.1 satisfies the Luce's Axiom, which states that the ratio of the probability of choosing one individual over an other is independent of the batch of individuals in which we choose. One implication of this axiom is "internal consistency", which results in the fact that a particular ranking of certain individuals does not depend on the set of individuals they arise from [17]. In our case, this means that we could extend our model by constraining an individual i to rank only a subset A of the network by just changing the terms in 1.1 by using $(\lambda_{i,j})_{j\in A}$ in the sums, (we could think of a student ranking his classmates, even if his best friend belongs to another class). Although we do not position ourselves in this context, this remark further justifies the use of this model for its simplicity.

In decision theory, this model is also equivalent to the use of the Stochastic Utility Model [7], that assumes rational choice behavior where the "utility" of an individual j for the individual i would be modeled with $\lambda_{i,j}$ and a random term.

Many methods have been developed to infer the parameters of this model. Hunter [17] derived MM algorithms, based on surrogates functions, to find the MLE (Maximum Likelihood Estimate) of the model, and proved the existence and uniqueness of the MLE if the network is strongly connected (where the edge $i \to j$ means that individual i is preferred over j by at least one individual). In a Bayesian configuration, [23] proposed a message passing algorithm that efficiently infer the parameters by approximating the posterior distribution. An efficient EM algorithm and Gibbs sampler have been developed in [27] by introducing latent variables in the model.

In addition to applying existing models that deal with covariates, this paper introduces a new representation of the model in terms of latent communities. This representation allows us to derive an EM algorithm and a Gibbs sampler to infer MAP estimates, as well as giving insight into potential community structure in the network. These statistical models are assessed on synthetic data and multiple applications.

Chapter 2

Community representation

2.1 Mathematical Formulation

In this section we suppose that the strength between individuals are characterized by their similitude to certain communities. We introduce p the number of latent communities in the network. For each individual i, let $w_i \in (\mathbb{R}^+)^p$, be the vector representation of i in the "community space". In this space, $w_{i,s}$ is the amplitude of affiliation of the individual i to the community s. We then model the strength of affiliation between individuals by the following formula:

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

For Bayesian inference, we also introduce the following priors, where $\Gamma(\alpha, \beta)$ is the gamma distribution:

$$w_{i,j} \sim \Gamma(a, b),\ a, b > 0 \tag{2.2}$$

We suppose here that the strength of the relations are only determined by the proximity of the individuals communities. Hence the relations in this case are reciprocal. From the reordering inequality, we can observe that, given the values attributed to the coordinates of w_i and w_j, $\lambda_{i,j}$ is maximized when both i and j have the same community ranking preferences. We can also rewrite:

$$\lambda_{i,j} = ||w_i||.||w_j||.cos(\theta_{i,j}) \tag{2.3}$$

We can interpret the norm of the vector as the individual global influence, and the $\theta_{i,j} \geq 0$ as the divergence in personality. Hence a given individual will appreciate another individual with great influence but different personality or views as much as an individual with low influence and with the same mindset.

We can also notice that:

$$\lambda = WW^T \tag{2.4}$$

Which is similar to the solution formulated in the domain of recommendation systems by collaborative filtering, where a matrix of data $M \in \mathbb{R}^{n \times p}$ is given (typically, $M_{i,j}$

represents the ranking of object j for individual i) and approximated by the product LU^T, where $L \in \mathbb{R}^{n\times k}$ and $U \in \mathbb{R}^{p\times k}$. LU^T become, then, the approximated score function between individual $i \in [n]$ and the "article" $j \in [p]$ [24].

2.2 Computation of the posterior

The log-posterior of the community representation is the sum of the log priors of the parameters and the log-likelihood:

$$l((w); D) = \sum_{i=1}^{n}\sum_{k=1}^{p}(a-1)ln(w_{i,k}) - bw_{i,k} + \sum_{i=1}^{n}\sum_{j=1}^{K} ln(\lambda_{i,\rho_j}) - ln(\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l}) \tag{2.5}$$

We can compute the gradient of the posterior according to the parameter $w_{r,s}$:
With

$$\frac{\partial \lambda_{i,j}}{\partial w_{r,s}} = 1_{i=r}w_{j,s} + 1_{j=r}w_{i,s} \tag{2.6}$$

We have :

$$\frac{\partial l((w); D)}{\partial w_{r,s}} = \frac{a-1}{w_{r,s}} - b + \sum_{j=1}^{K}\frac{1}{\lambda_{r,\rho_j^{(r)}}}w_{\rho_j^{(r)},s} + \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\lambda_{i,r}}w_{i,s}1_{\rho_j^{(r)}=r} - \\ \sum_{j=1}^{K}\frac{1}{\sum_{j\neq r}\lambda_{r,j} - \sum_{l=1}^{j-1}\lambda_{r,\rho_l^{(r)}}}\sum_{j\neq r}w_{j,s} - \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l^{(i)}}}w_{i,s} + \\ \sum_{j=1}^{K}\frac{1}{\sum_{j\neq r}\lambda_{r,j} - \sum_{l=1}^{j-1}\lambda_{r,\rho_l^{(r)}}}\sum_{l=1}^{j-1}w_{\rho_l^{(r)},s} + \sum_{i=1;i\neq r}^{n}\sum_{j=1}^{K}\frac{1}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{j-1}\lambda_{i,\rho_l^{(i)}}}\sum_{l=1}^{j-1}1_{\rho_l^{(i)}=r}w_{i,s} \tag{2.7}$$

The gradient is too complex to derive a formula for a MAP directly, hence justifying the introduction of latent variables.

2.3 EM algorithm with community representation

The posterior distribution over the (w) derived in equation 2.7 is somewhat very complicated. Hence, in this section, we introduce latent variables in order to derive an EM algorithm.

2.3.1 Mathematical formulation

We recall the likelihood of the Plackett-Luce Model:

$$p(\rho_i|(\lambda)) = \prod_{k=1}^{K}\frac{\lambda_{i\rho_k}}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l}} \tag{2.8}$$

With:

$$\lambda_{i,j} = \sum_{k=1}^{p} w_{i,k} w_{j,k} \tag{2.9}$$

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

$$L(\lambda, Z) = P(D, Z|\lambda) = (\prod_{i=1}^{n}\prod_{k=1}^{K} \frac{\lambda_{i\rho_k}}{\sum_{j\neq i}\lambda_{i,j} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l}})P(Z|D,\lambda) \tag{2.10}$$

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

We first introduce, where $\mathcal{E}(\lambda)$ is the exponential distribution:

$$V_{i,j,k} \sim \mathcal{E}(w_{i,k}w_{j,k}),\ 1 \leq i \neq j \leq n,\ \ 1 \leq k \leq p \tag{2.11}$$

$$V_{i,j} = min(V_{i,j,1}, ..., V_{i,j,p}) \sim \mathcal{E}(\lambda_{i,j}) \tag{2.12}$$

And

$$Z_{i,k}|\rho_i = min(V_{i,j})_{j\neq i,\rho_i[1],...,\rho_i[k-1]} \sim \mathcal{E}(\sum_{j\neq i,\rho_i[1],...,\rho_i[k-1]} \lambda_{i,j}), 1 \leq k \leq K \tag{2.13}$$

An interpretation of these latent variables is that the next person ranked by an individual is equivalent to the first arrived individual in a waiting process (among those remaining to be ranked) following an average waiting time of $\frac{1}{\lambda_{i,j}}$. The arriving time of this same individual is equivalent to the first arriving time of p other individuals in a waiting process following an average waiting time of $\frac{1}{w_{i,k}w_{j,k}}$ which is the contribution of the k^{th} community.

This latent variable allows us to suppress the denominator in 2.10. To suppress the nominator we introduce the following latent variables, where $Cat(p_1, ..., p_k)$ is the categorical distribution:

$$Y_{i,j} \sim Cat(\frac{w_{i,1}w_{\rho_i[j],1}}{\lambda_{i,\rho_i[j]}}, ..., \frac{w_{i,p}w_{\rho_i[j],p}}{\lambda_{i,\rho_i[j]}}), 1 \leq j \leq K \tag{2.14}$$

These variables can be interpreted as the community responsible for the choice of a certain individual.

We obtain the following log-likelihood distribution:

$$\mathcal{L}((w); Y, Z, D) = \sum_{i=1}^{n}\sum_{k=1}^{K}\left\{ -(\sum_{l\neq i}\lambda_{i,l} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l})z_{i,k} + \sum_{l=1}^{p}\delta_{y_{i,k},l}(ln(w_{i,l}) + ln(w_{\rho_k,l}))) \right\} \tag{2.15}$$

2.3.2 EM formulation

E-step:

$$\mathbb{E}_{Z,Y|\rho,(w)^*}(\mathcal{L}((w);Y,Z,D)) =$$
$$\sum_{i=1}^{n}\sum_{k=1}^{K}\Bigg\{-\frac{(\sum_{l\neq i}\lambda_{i,l}-\sum_{l=1}^{k-1}\lambda_{i,\rho_l})}{(\sum_{l\neq i}\lambda^*_{i,l}-\sum_{l=1}^{k-1}\lambda^*_{i,\rho_l})}+\sum_{l=1}^{l}\frac{w^*_{i,l}w^*_{\rho_k,l}}{\lambda^*_{i\rho_k}}(ln(w_{i,l})+ln(w_{\rho_k,l})))\Bigg\} \tag{2.16}$$

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

$$Q(w|w^*) = \sum_{i=1}^{n}\Bigg[\sum_{k=1}^{K}\Bigg\{-\frac{(\sum_{l\neq i}\lambda_{i,l}-\sum_{l=1}^{k-1}\lambda_{i,\rho_l})}{(\sum_{l\neq i}\lambda^*_{i,l}-\sum_{l=1}^{k-1}\lambda^*_{i,\rho_l})}+$$
$$\sum_{l=1}^{p}\frac{w^*_{i,l}w^*_{\rho_k,l}}{\lambda^*_{i,\rho_k}}(ln(w_{i,l})+ln(w_{\rho_k,l})))\Bigg\}+\sum_{l=1}^{p}(a-1)ln(w_{i,l})-bw_{i,l}\Bigg] \tag{2.17}$$

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

$$\frac{\partial\lambda_{i,j}}{\partial w_{r,s}} = 1_{i=r}w_{j,s}+1_{j=r}w_{i,s} \tag{2.18}$$

We obtain this time, after computation:

$$w_{r,s}\frac{\partial Q}{\partial w_{r,s}} = C(r,s,w^*)-A(r,s,w,w^*)w_{r,s} \tag{2.19}$$

With

$$C(r,s,w^*) = a-1+\sum_{k=1}^{K}\frac{w^*_{r,s}w^*_{\rho_r[k],s}}{\lambda^*_{r,\rho_r[k]}}+\sum_{i\neq r}\frac{w^*_{i,s}w^*_{r,s}}{\lambda^*_{i,r}}\mathbb{1}(r\in\rho_i) \tag{2.20}$$

and

$$A(r,s,w,w^*) = b\ +\ \Bigg(\sum_{k=1}^{K}\frac{1}{\Lambda^*(r,k)}\Bigg)\Bigg(\sum_{l\neq r}w_{l,s}\Bigg)+\sum_{i\neq r}w_{i,s}\Bigg(\sum_{k=1}^{K}\frac{1}{\Lambda^*(i,k)}\Bigg)-$$
$$\sum_{l=1}^{K-1}w_{\rho_r(l),s}\Bigg(\sum_{k=l+1}^{K}\frac{1}{\Lambda^*(r,k)}\Bigg)-\sum_{i\neq r}w_{i,s}\Bigg(\sum_{k>rank_{\rho_i}(r)}^{K}\frac{1}{\Lambda^*(i,k)}\Bigg)\mathbb{1}(r\in\rho_i) \tag{2.21}$$

where

$$\Lambda^*(i,k) = \sum_{l\neq i}\lambda^*_{i,l}-\sum_{l=1}^{k-1}\lambda^*_{i,\rho_i[l]} \tag{2.22}$$

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 check that A and C are always positive if $a \geq 1$. 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 increase the posterior likelihood. We can update iteratively $w_{r,s}$ with

$$w^{(t)}_{r,s} = \frac{C(r,s,w^{(t-1)})}{A(r,s,w^{(t-1)})} \tag{2.23}$$

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

2.3.3 EM formulation with variable ranks

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

$$\mathcal{L}((w); Y, Z, D) = \sum_{i=1}^{n}\sum_{k=1}^{K_i}\left\{ -(\sum_{l\neq i}\lambda_{i,l} - \sum_{l=1}^{k-1}\lambda_{i,\rho_l})z_{i,k} + \sum_{l=1}^{p}\delta_{y_{i,k},l}(ln(w_{i,l}) + ln(w_{\rho_k,l})))\right\} \tag{2.24}$$

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:

$$w^{(t)}_{r,s} = \frac{C(r,s,w^{(t-1)})}{A(r,s,w^{(t-1)})} \tag{2.25}$$

with

$$C(r,s,w^*) = a - 1 + \sum_{k=1}^{K_r}\frac{w^*_{r,s}w^*_{\rho_r[k],s}}{\lambda^*_{r,\rho_r[k]}} + \sum_{i\neq r}\frac{w^*_{i,s}w^*_{r,s}}{\lambda^*_{i,r}}\mathbb{1}(r\in\rho_i) \tag{2.26}$$

and

$$\begin{aligned} A(r,s,w,w^*) = b \ + \ & \left(\sum_{k=1}^{K_r}\frac{1}{\Lambda^*(r,k)}\right)\left(\sum_{l\neq r}w_{l,s}\right) + \sum_{i\neq r}w_{i,s}\left(\sum_{k=1}^{K_i}\frac{1}{\Lambda^*(i,k)}\right) - \\ & \sum_{l=1}^{K_r-1}w_{\rho_r(l),s}\left(\sum_{k=l+1}^{K_r}\frac{1}{\Lambda^*(r,k)}\right) - \sum_{i\neq r}w_{i,s}\left(\sum_{k>rank_{\rho_i}(r)}\frac{1}{\Lambda^*(i,k)}\right)\mathbb{1}(r\in\rho_i) \end{aligned} \tag{2.27}$$

where

$$\Lambda^*(i,k) = \sum_{l\neq i}\lambda^*_{i,l} - \sum_{l=1}^{k-1}\lambda^*_{i,\rho_i[l]} \tag{2.28}$$

2.3.4 Handling missing data

If a certain individual has not given any ranking of his friends, the EM algorithm with variable ranks handles this situation. In this case, Eq. 2.25 remains the same for the missing data, Eq. 2.26 becomes :

$$C(r,s,w^*) = a - 1 + \sum_{i \neq r} \frac{w^*_{i,s} w^*_{r,s}}{\lambda^*_{i,r}} \mathbb{1}(r \in \rho_i) \tag{2.29}$$

and 2.27:

$$A(r,s,w,w^*) = b + \sum_{i \neq r} w_{i,s} \left(\sum_{k=1}^{K_i} \frac{1}{\Lambda^*(i,k)} \right) - \sum_{i \neq r} w_{i,s} \left(\sum_{k > rank_{\rho_i}(r)}^{K_i} \frac{1}{\Lambda^*(i,k)} \right) \mathbb{1}(r \in \rho_i) \tag{2.30}$$

We can then predict the ranks of the individuals with regard to this person that has not expressed himself by sampling from the Plackett-Luce model with the optimized parameters.

2.4 Degree Correction

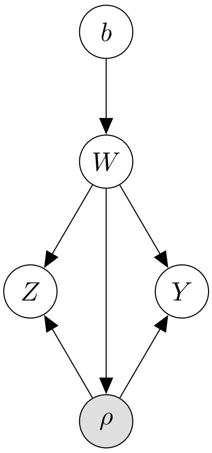


Figure 2.1: Scheme Latent Variables

In this section we refine the Plackett-Luce model to study the potential global influence of a community over the others, to do this, we introduce the parameters $(b_k)_{k \in \{1,\ldots,p\}} > 0$, such that:

$$\forall i \in [n], w_{i,k} \sim \Gamma(a, b_k) \tag{2.31}$$

We use the same approach as in the EM algorithm, we introduce the same Latent variables to represent the model. We consider the general case in which the number of ranks is not fixed for each individuals. We use the conjugate prior of the gamma distribution $\Gamma(a, b_k)$ for the prior of b_k, namely:

$$b_k \sim \Gamma(\alpha_0, \beta_0) \tag{2.32}$$

The new set of variables is resumed in Figure 2.1.

Using these latent variables, we can compute the conditional distribution of every variable and build a Gibbs Sampler from which we can sample from the full posterior distribution. We update iteratively b, Y, W, z, as follows at iteration t:

1. $b_s^{(t)} \mid W^{(t-1)} \sim \Gamma(\alpha_0 + na, \beta_0 + \sum_{i=1}^{n} w_{i,s}^{(t-1)})$

2. $Y_{i,j}^t \mid W^{(t-1)} \sim Cat(\frac{w_{i,1}^{(t-1)} w_{\rho_i[j],1}^{(t-1)}}{\lambda_{i,\rho_i[j]}^{(t-1)}}, ..., \frac{w_{i,p}^{(t-1)} w_{\rho_i[j],p}^{(t-1)}}{\lambda_{i,\rho_i[j]}^{(t-1)}})$

3. $Z_{i,k}^{(t)} | W^{(t-1)}, \rho_i \sim \mathcal{E}(\sum_{j \neq i, \rho_i[1], ..., \rho_i[k-1]} \lambda_{i,j}^{(t-1)}), 1 \leq k \leq K_i$

4. $w_{r,s}^{(t)} \mid b^{(t)}, Y^{(t)}, Z^{(t)}, W_{\setminus\{r,s\}}^{(t)}, \rho \sim \Gamma\Big(a - 1 + \sum_{1 \leq k \leq K_r} (\delta_s(y_{r,k})) + \sum_{i:i \to r} (\delta_s(y_{i,rank_r(\rho_i)}));$
$b_s^{(t)} + \big(\sum_{k=1}^{K_r} z_{r,k}(\sum_{l \neq r} w_{l,s} - \sum_{l=1}^{k-1} w_{\rho_r[l],s})\big) + \big(\sum_{i \neq r} \sum_{k=1}^{K_i} w_{i,s} z_{i,k} \mathbb{1}\{r \notin (\rho_i[1], ..., \rho_i[k-1])\}\big)\Big)$

The log-posterior distribution is given by:

$$log(P(W, (b)|D)) = (\alpha_0 - 1) \sum_{s=1}^{p} log(b_s) + (nlog(a) - \beta_0) \sum_{s=1}^{p} b_s +$$
$$\sum_{i=1}^{n} \sum_{s=1}^{p} (a-1) log(w_{i,s}) - b_s w_{i,s} + \sum_{i=1}^{n} \sum_{j=1}^{K_i} log(\lambda_{i,\rho_j}) - log(\sum_{j \neq i} \lambda_{i,j} - \sum_{l=1}^{j-1} \lambda_{i,\rho_l}) \tag{2.33}$$

2.5 Identifiability

In our model, the prior as well as the likelihood are invariant by relabelling, hence, if one wants to use a MCMC approach to estimate the parameters, a posterior mean estimate would be completely inappropriate. Richardson [14] discusses this issue and proposes to incorporate "artificial" constraints in the prior to make the labelling identifiable. Constraints could be the addition of a term $\mathbb{1}(b_1 < b_1 < ... < b_p)$ in the prior of the model. In our case, the result of the EM algorithm is deterministic and characterized by the initialization. In the case of our Gibbs Sampler, we decided to keep the original priors, but we took maximum a posteriori estimates of the parameters instead of the posterior mean. We compared these estimates with the resulting posterior distributions.

An other remark is the fact that the norms of the vectors $(\lambda_i)_{i \in [n]}$ are not likelihood identifiable. However, the representation we have adopted in terms of latent communities does not allow us to derive a simple reparameterization of the $w_{i,k}$ to take advantage of this property.

2.6 Model Selection

Our representation with latent communities introduces the parameter P, the number of communities in the network. A different value of P involves a different model with a different number of parameters. It would be interesting to infer P from the data. Since the computation $p(P|D)$ involves intractable integrals, the use of simulations seems inevitable.

2.6.1 Reversible Jump MCMC

A first approach for model selection would be to treat P as parameter and use a Markov Chain Monte Carlo method to sample from the full posterior distribution. However, a difficulty arises when we have to compute the acceptance ratio for the changement of state. Indeed, since the models don't have the same number of parameters, the respective densities are not comparable. The use of Reversible Jump MCMC (RJMCMC), would, given a prior over the models, solve this problem. The MAP of P could, then, be estimated by its most encountered value in the Markov Chain. We have no place to justify the use of this method, however, a full development of the RJMCMC can be found in [12]. See the appendix for a description of the algorithm. Here is one possible formulation of the algorithm for our purpose:

Our parameters are $w_{i,k}$ with $i \in \{1, ..., n\}$, $k \in \{1, ..., p\}$ for a model $P = p$ with p latent communities. We fix the number P_{max} of models we will test, so that $1 \leq P \leq P_{max}$.
Suppose the state is $X_t = (w^{(t)}_{p^{(t)}}, p^{(t)})$, to get irreducibility, we need both fixed and variable dimension moves. Here we suppose that the user has already elicited a prior over the models $\pi(p)$, depending on the case studied:

1. Propose $p^* = p^{(t)}$ with probability $\frac{1}{2}$, or propose $p^* = p' \in [P_{max}]$ with probability $\frac{1}{2P_{max}}$.

2. (a) If $p^* = p^{(t)}$, then propose $w^*_{i,j} \sim \Gamma(\alpha = 2, \beta = \frac{1}{w^{(t)}_{i,j}})$, for $(i,j) \in [n] \times [p^{(t)}]$.
   Accept with probability
   $\alpha(X^{(t)}, X^*) = min(1, \frac{P(w^*, p^*|D) \prod_{i,j} \Gamma(w^{(t)}_{i,j}, \alpha=2, \beta=\frac{1}{w^*_{i,j}})}{P(w^{(t)}, m^{(t)}|D) \prod_{i,j} \Gamma(w^*_{i,j}, \alpha=2, \beta=\frac{1}{w^{(t)}_{i,j}})})$.
   Otherwise reject and keep $X^{(t+1)} = X^{(t)}$.

   (b) If $p^* > p^{(t)}$, then propose $w^*_{i,j} \sim \Gamma(\alpha = a, \beta = b)$ for $i \in \{1, ..., n\}$ and $j \in \{p^{(t)} + 1, ..., p^*\}$ and accept with probability
   $\alpha(X^{(t)}, X^*) = min(1, \frac{P(w^*, m^*|D)}{P(w^{(t)}, m^{(t)}|D) \prod_{i,j \in \{m^{(t)}+1, ..., m^*\}} \Gamma(w^{(t)}_{i,j}, \alpha=a, \beta=b)})$.
   Otherwise, reject the proposal and keep $X^{(t+1)} = X^{(t)}$.

   (c) If $p^* < p^{(t)}$, then propose w^* by dropping the parameters $w^{(t)}_{i,j}$ for $i \in \{1, ..., n\}$ and $j \in \{p^* + 1, ..., p^{(t)}\}$ and accept with probability
   $\alpha(X^{(t)}, X^*) = min(1, \frac{P(w^*, p^*|D) \prod_{i,j \in \{p^{(t)}+1, ..., p^*\}} \Gamma(w^{(t)}_{i,j}, \alpha=a, \beta=b)}{P(w^{(t)}, p^{(t)}|D)})$.
   Otherwise, reject the proposal and keep $X^{(t+1)} = X^{(t)}$.

This version is a possible algorithm one can implement to perform model selection. However there are some precautions to take and multiple limitations of this method: First, since the different models that we propose are "refinements" of the representation of the data, in the sense that a higher number of parameters only increases the precision with which we can describe the data, the role played by the prior on the models is crucial. The elicitation of the prior will determine a "regularization" error. For instance, if one chooses a uniform prior, we can expect the chain to indicate a preference for the model with the highest number of parameters P_{max}. Second, the reversibility of the chain implies the conception of a diffeomorphism $\Psi_{p_1 \to p_2}$ between models of different dimension. When, for instance, the algorithm jumps from a model of dimension p_1 to p_2 where $p_2 > p_1$, the literature often suggests sampling the additional parameters either from the prior directly [21] or the posterior density of the additional parameters given the kept parameters [16]. In our case the posterior is not computable unless introducing additional latent variables (see Section on the Degree Correction). Moreover, one can notice the moves with fixed dimensions are equivalent to a classic M-H exploration of the space of parameters. From these remarks, we can notice that the jumps with variable dimension do not take into account the past explorations of the parameter space of each model. To cope with this issue, we could think of storing the previous explorations of each model by the constant dimension moves, and take these past explorations into account when defining a proposal for the jumps. However, this would imply that the diffeomorphisms defining the jumps between models is not longer homogeneous, which is a case not treated in [12]. Hence, we can expect this algorithm to perform poorly in our case, especially when n increases, since the difference in dimension between models become wider, and so the exploration in each model more crucial.

2.6.2 Information Criterion

A second approach of model selection comes from information theory [11], [25]. When fitting the data, a more refined model with additional parameters will mimic the data more accurately, which can result in overfitting the data. The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are estimators that try to balance the accuracy of the representation and the flexibility of a model with a penalty term on its complexity. These estimators are given by:

$$AIC = 2k - 2log(L(\hat{\theta}_{MLE})) \tag{2.34}$$

$$BIC = 2log(n)k - 2log(L(\hat{\theta}_{MLE})) \tag{2.35}$$

Where k is the number of parameters in the model, in our case, $k = n \times p$, and $\hat{\theta}_{MLE}$ is the maximum likelihood estimate. A model will then be selected to minimize one of these criteria.

Chapter 3

Model with Covariates

3.1 Covariate models

The modelling of social network data has been carried out from various perspectives. In particular, the use of Exponential Random Graph Models (ERGM) has been thoroughly studied in this objective [28]. However, these models are computationally expensive, are exposed to the problem of confounding nodal covariate effects with nodal degree effects and are limited to binary relational data (Pavel N. Krivitsky [35] has recently refined an ERGM model to incorporate Rank-Order relational data). In this chapter we develop a natural extension of the Plackett-Luce model that deals with covariates as well as a regression model with a different likelihood that deals with the previously mentioned limitations of the ERGMs models in the context of rank nomination networks.

3.2 The Plackett-Luce Model with Covariates

3.2.1 Formulation

A natural refinement of the Plackett-Luce model to incorporate covariates is by adopting the following model for the parameters λ:

$$log(\lambda_{i,j}) = \boldsymbol{\beta^T x_{i,j}} \tag{3.1}$$

Where $x_{i,j}$ represents the covariates of the relation between individual i and j. Since the Plackett-Luce model is defined up to a multiplicative constant, parameters corresponding to intrinsic characteristics of individual i are not identifiable. The covariates $\boldsymbol{x_{i,j}}$ also include the variables $(\mathbb{1}\{l = j\})_{l\in[n]}$, the corresponding parameter β_l representing the popularity of individual l. We can put a prior on the parameters (β):

$$p(\beta_i) \sim \mathcal{N}(0, \sigma^2) \tag{3.2}$$

Then, we obtain the following log posterior:

$$\mathcal{L}(\beta|D) \equiv -\frac{\beta^T\beta}{2\sigma^2} + \sum_{i=1}^{N}\sum_{j=1}^{K_i} \beta^T x_{i,\rho_i[j]} - log\Big(\sum_{l\neq i,\rho_i[1],\dots,\rho_i[j-1]} exp(\beta^T x_{i,l})\Big) \tag{3.3}$$

3.2.2 Fitting the model

In order to infer the parameters, we can simply implement a metropolis-hasting algorithm by designing a proposal distribution and compute point estimates from the approximated posterior distribution. Another method consists in using an MM algorithm similar to [17] or [26] used in the context of the Bradley-Terry model or a mixture of Plackett-Luce.

In equation 3.3, we cannot optimize directly the objective function because of the last term $-log\Big(\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\beta^T x_{i,l})\Big)$. However, $-log(\theta)$ is convex, hence, with $g(\theta) \geq g(\bar{\theta}) + g'(\bar{\theta})(\theta - \bar{\theta})$ for a convex function we have:

$$\begin{aligned} -log\Big(\sum_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\beta^T x_{i,l})\Big) \geq & -log\Big(\sum_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\bar{\beta}^T x_{i,l})\Big) \\ & + 1 - \frac{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\beta^T x_{i,l})}{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\bar{\beta}^T x_{i,l})} \end{aligned} \quad (3.4)$$

Where $\bar{\beta}$ is a constant value of β. We obtain, without the constants:

$$\mathcal{L}(\beta|D) \geq -\frac{\beta^T\beta}{2\sigma^2} + \sum_{i=1}^{N}\sum_{j=1}^{K_i} \beta^T x_{i,\rho_i[j]} - \frac{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\beta^T x_{i,l})}{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\bar{\beta}^T x_{i,l})} \quad (3.5)$$

The last term is still an issue, however, $\beta \rightarrow -exp(\beta^T x_{i,l})$ is concave. With $g(\theta) \geq g(\bar{\theta}) + \{g'(\bar{\theta})\}^T(\theta - \bar{\theta}) + \frac{1}{2}(\theta - \bar{\theta})^T \boldsymbol{B}(\theta - \bar{\theta})$ where $\boldsymbol{B}$ is negative definite and $\boldsymbol{B} < \frac{\partial^2 g(\bar{\theta})}{\partial \theta^2}$, see [9] for quadratic surrogates:

$$-exp(\beta^T x_{i,l}) \geq -exp(\bar{\beta}^T x_{i,l}) - x_{i,l}^T exp(\bar{\beta}^T x_{i,l})(\beta - \bar{\beta}) - \frac{1}{2}(\beta - \bar{\beta})^T \boldsymbol{B}(\beta - \bar{\beta}) \quad (3.6)$$

Where $\boldsymbol{B} = x_{i,l}x_{i,l}^T$, hence we obtain, without the constants:

$$\begin{aligned} \mathcal{L}(\beta|D) \geq & -\frac{\beta^T\beta}{2\sigma^2} + \sum_{i=1}^{N}\sum_{j=1}^{K_i} \beta^T x_{i,\rho_i[j]} \\ & - \frac{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} x_{i,l}^T exp(\bar{\beta}^T x_{i,l})\beta + \frac{1}{2}\beta^T(x_{i,l}x_{i,l}^T)\beta - \beta^T(x_{i,l}x_{i,l}^T)\bar{\beta}}{\sum\limits_{l\neq i,\rho_i[1],\ldots,\rho_i[j-1]} exp(\bar{\beta}^T x_{i,l})} \end{aligned} \quad (3.7)$$

We can optimize the second term directly since we have a formula with quadratic terms in β, Computing the optimized value β requires to inverse a matrix of the size of the number of covariates. We obtain :

$$\beta = \Bigg[\sum_{i=1}^{N} \sum_{j=1}^{K_i} \frac{\sum\limits_{l \neq i, \rho_i[1], \ldots, \rho_i[j-1]} x_{i,l} x_{i,l}^T}{\sum\limits_{l \neq i, \rho_i[1], \ldots, \rho_i[j-1]} exp(\bar{\beta} x_{i,l})} + \frac{1}{\sigma^2} I \Bigg]^{-1} \times$$

$$\Bigg[\sum_{i=1}^{N} \sum_{j=1}^{K_i} x_{i,\rho_i[j]} - \frac{\sum\limits_{l \neq i, \rho_i[1], \ldots, \rho_i[j-1]} exp(\bar{\beta}^T x_{i,l}) x_{i,l} - (x_{i,l} x_{i,l}^T) \bar{\beta}}{\sum\limits_{l \neq i, \rho_i[1], \ldots, \rho_i[j-1]} exp(\bar{\beta}^T x_{i,l})} \Bigg] \tag{3.8}$$

The matrix inversion is licit since it is a positive definite matrix.

3.3 The social relations regression model (SRRM) Model

One drawback of the previous model is that it cannot capture the sociability of the individuals, or their predisposition to have positive relationship with numerous other individuals. This can be interpreted by the fact that any row effect in the previous model vanishes in the Plackett-Luce model, and that the number of rankings is fixed in advance and independent of the values of the $\lambda_{i,j}$. This difficulty is inherent to the Plackett-Luce likelihood, where the number of ranking for each individual is not linked to the values of the $\lambda_{i,j}$. We can change the formulation of the problem to bypass this issue by adopting the model used by P. Hoff [31].

3.3.1 Formulation

We now authorize the matrix representing the strength of affiliation between individuals λ to be negative. We then introduce **S** the sociomatrix given in data.
$\mathbf{S} = \{s_{i,j} : i \neq j\}$, is 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 $d_i \leq m$. Now instead of using the Plackett-Luce model that generated **S** with λ, we create the constraints:

$$s_{i,j} = [(m - rank_i(\lambda_{i,j} + 1) \wedge 0)] \times 1(\lambda_{i,j} > 0) \tag{3.9}$$

and its inverse:

$$s_{i,j} > 0 \Rightarrow \lambda_{i,j} > 0 \tag{3.10}$$

$$s_{i,j} > s_{i,k} \Rightarrow \lambda_{i,j} > \lambda_{i,k} \tag{3.11}$$

$$s_{i,j} = 0 \text{ and } d_i < m \Rightarrow \lambda_{i,j} \leq 0 \tag{3.12}$$

Now if a statistical model is formulated for $\{p(\lambda|\theta) : \theta \in \Theta\}$, inference for the parameter θ can be derived from the observed scores $\mathbf{S}$ through the likelihood:

$$L_F(\theta : \mathbf{S}) = Pr(\lambda \in F(S)|\theta) = \int_{F(S)} p(\lambda|\theta) d\mu(\lambda) \tag{3.13}$$

Where $F(S)$ denote the set of λ-values that are consistent with S in terms of the three previous equations, and μ is a measure that dominates the probability densities $\{p(\lambda|\theta) : \theta \in \Theta\}$.

If the statistical model allows it conveniently, we can formulate a Gibbs Sampler from this process : Given current values of (θ, λ), one step of the Gibbs sampler proceeds by updating the values as follows:

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

This process will generate values of $\lambda_{i,j}$ from its full distribution, constrained to the conditions dictated by the matrix $\mathbf{S}$. If a Gibbs Sampler cannot be formulated, a Metropolis-Hasting algorithm can be developed with the appropriate proposal distribution.

3.3.2 Regression Model

Given this process for treating ranked data, we choose, as in [31], to take the following regression model, called the Social Relations Regression Model (SRRM):

$$y_{i,j} = \boldsymbol{\beta^T x_{i,j}} + a_i + b_j + \epsilon_{i,j} \tag{3.14}$$

$$\begin{pmatrix} a_i \\ b_i \end{pmatrix}, i = 1, ..., n \sim \text{i.i.d normal}(0, \Sigma_{ab}) \tag{3.15}$$

$$\begin{pmatrix} \epsilon_{i,j} \\ \epsilon_{j,i} \end{pmatrix}, i = 1, ..., n \sim \text{i.i.d normal}(0, \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}) \tag{3.16}$$

The additive row effect a_i can be interpreted as i's "Sociability" whereas the additive column effect b_i can be interpreted as i's "Popularity.", the introduction of these parameters account for the disparity in in-degree or out-degree in the network. The parameter ρ represents potential correlation between $y_{i,j}$ and $y_{j,i}$. The covariance matrix Σ_{ab} represents the correlation between the sociability and popularity of an individual [6].

Chapter 4

Experimental Results

4.1 EM Algorithm

In this section we evaluate the performance and the results obtained with the EM algorithm previously described in the context of Community representation. We implemented the algorithm in R, and made it able to support variable ranks and missing data, hence any type of networks as input.

4.1.1 Synthetic Data

Convergence

We tested our EM algorithm on a simulated random network of size $n = 20, K = 5, p = 4, a = 2, b = 2$ and plotted the log-posterior at each single 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) randomly drawn from the priors.

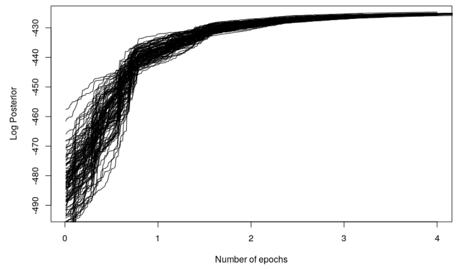


Figure 4.1: Performance on the first 4 epochs

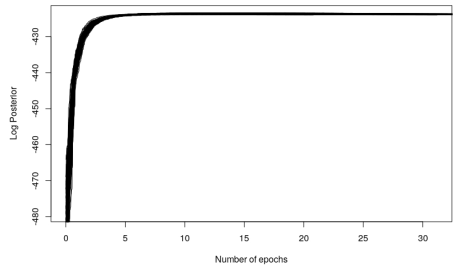


Figure 4.2: Performance on 30 epochs

From these figures we can draw several remarks:

1. The algorithm converges to a local maximum of the posterior.

2. The convergence is fast in this example: in two or three epochs we already have a good approximation of the MAP.
3. No matter the initial value of (w), the algorithm seems to converge to the same value of the maximum of the posterior. Hence, suggesting that the maximum found is the global maximum (not unique, at least because of the labelling of the communities).

Performance

Here we randomly drew graphs of size $n = 20$ and performed the EM algorithm as well as the L-BFGS-B optimization algorithm [8]. In order to compare their performances at finding the MAP, 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.

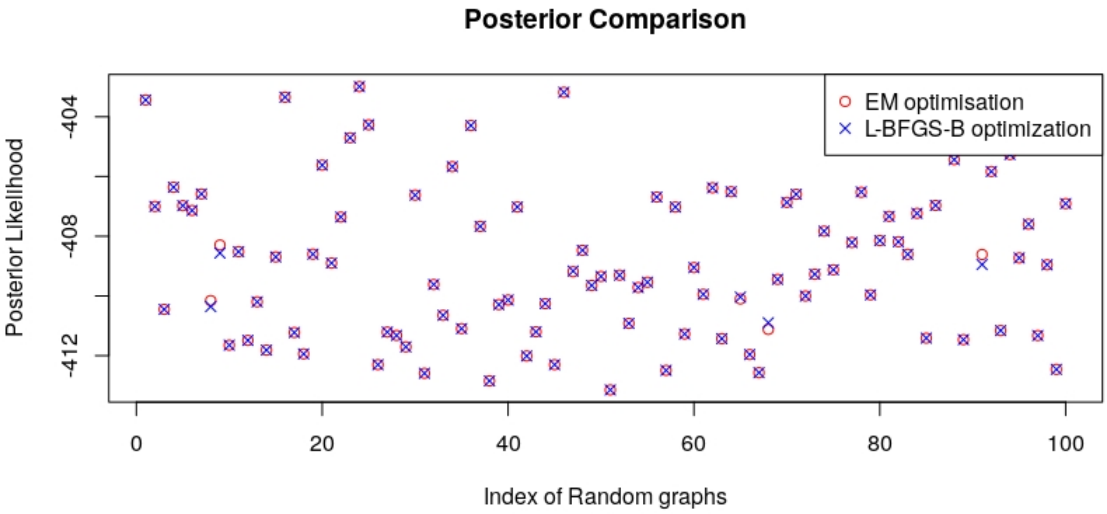


Figure 4.3: Performance Comparison between the EM and L-BFGS-B algorithm

.

We can see no significant differences between the performance at finding the MAP between a state-of-the art optimization algorithm and our EM algorithm. In terms of efficiency, our EM algorithm is approximately ten times less efficient, but the implementation is not optimized compared to an algorithm available in a library.

Community Detection

We assessed the performance of our model at performing community detection. To do so, we applied our algorithm to random directed graphs drew from the Stochastic Block Model (SBM) [29]. The SBM is a generative model for random graphs that is a generalization of the Erdős-Rényi model where every edge has a constant probability

p to be created. The SBM differs by allowing different probabilities $p_{i,j}$ inside and between "blocks" of nodes. Here is a proper formulation:

1. Let n be the number of vertices of the network.

2. Let $C_1, ..., C_r$ be a partition of the n nodes into r communities.

3. Let P be a $r \times r$ symmetric matrix of probability.

Then, a random graph is sampled from this model by sampling the edge set as follows: For all $u \in C_i$ and $v \in C_j$, u and v are connected with an edge with probability $P_{i,j}$.

Here we took two balanced blocks for n nodes, defined with the inter- and intra-block probability $\frac{c_{out}}{n}$ and $\frac{c_{in}}{n}$. We then assessed the quality of a partition by computing the "recovery" which is the proportion of nodes rightly attributed to its true block. We compared the results of our algorithm with a bi-partitioning spectral algorithm based on the leading eigenvector of the Laplacian matrix of the graph [32], which is equivalent to modularity optimization. We took graphs of size $n = 100$ with parameters $p = 2, a = 2, b = 1$ and averaged the results of each value of c_{in} and c_{out} with 35 simulations. From the literature [33], we know that when $c_{in} - c_{out} \leq \sqrt{2(c_{in} + c_{out})}$ no algorithm perform better than random. In the following Figure 4.4, we took the average degree $c = 3$, and computed the recovery for different values of c_{in} and c_{out}. In this case, the theoretical value of $\frac{c_{in} - c_{out}}{2}$ from which community detection is possible is $c_{lim} = \sqrt{3} \approx 1.7$.

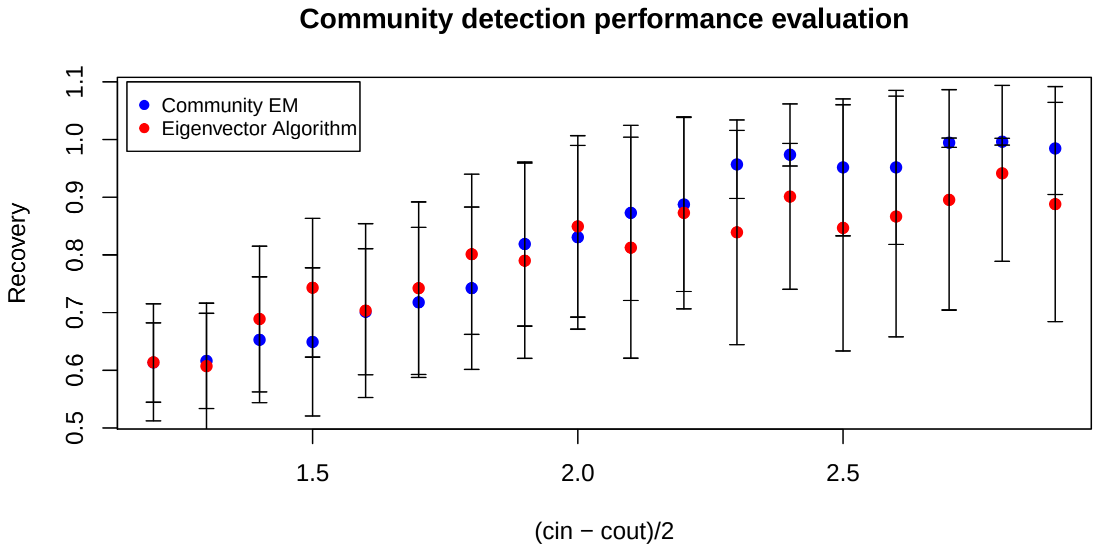


Figure 4.4: Community Detection performance comparison

From the results, Figure 4.4, we can see that our model perform similarly or slightly better that an algorithm based on modularity optimization.

Prediction

Here we assess the ability of our model to predict preferred individuals. We proceeded as follows:

1. We fixed the size of the networks $n = 40$, the parameters for graph generation: $a = 2$, $b = 1$, $p = 6$.
2. We fixed the number K_{max} of individuals we will study for the assessment.
3. Then, for each $K \in \{1, ..., K_{max} - 1\}$:
   (a) We generated a network where each individual ranks their K preferred other individuals.
   (b) We inferred the parameters $\lambda_{inferred}$ from this network with the EM algorithm, and computed the $K_{max} - K$ preferred individuals for each node, after having removed the one already selected in the original network.
   (c) We compared this list with the real list of preferred individuals drawn from the original λ. For the comparison of two partitions of size l, we computed the mean rank distance of each element of the lists:
   For ρ_1 and ρ_2 two partitions of $[n]$:
   $$Err(\rho_1, \rho_2) = \frac{1}{n}\sum_{i=1}^{n} |i - s_{1,2}(i)| \tag{4.1}$$
   Where $s_{1,2}(i)$ is the index of $\rho_1(i)$ in ρ_2.
   (d) We repeated the previous three points $n_{simu} = 30$ times and averaged the errors obtained.

We compared the errors obtained with errors we would obtain with a totally random strategy of prediction. In particular we have after computation:

$$\frac{1}{n}\mathbb{E}_{\sigma_n}[\sum_{k=1}^{n} |\sigma(k) - k|] = \frac{n^2 - 1}{3n} \tag{4.2}$$

And:

$$\frac{1}{T}\mathbb{E}_{\sigma_n}[\sum_{k=1}^{T} |\sigma(k) - k|] = \frac{3(n+1)(n-T-1) + (T+1)(2T+1)}{6n} \tag{4.3}$$

which is equal to the previous formula for $T = n$. In particular, we plotted the errors with the random errors, computed by the previous formula with $n_{formula} = n_{network} - K$ and $T = K_{max} - K$.

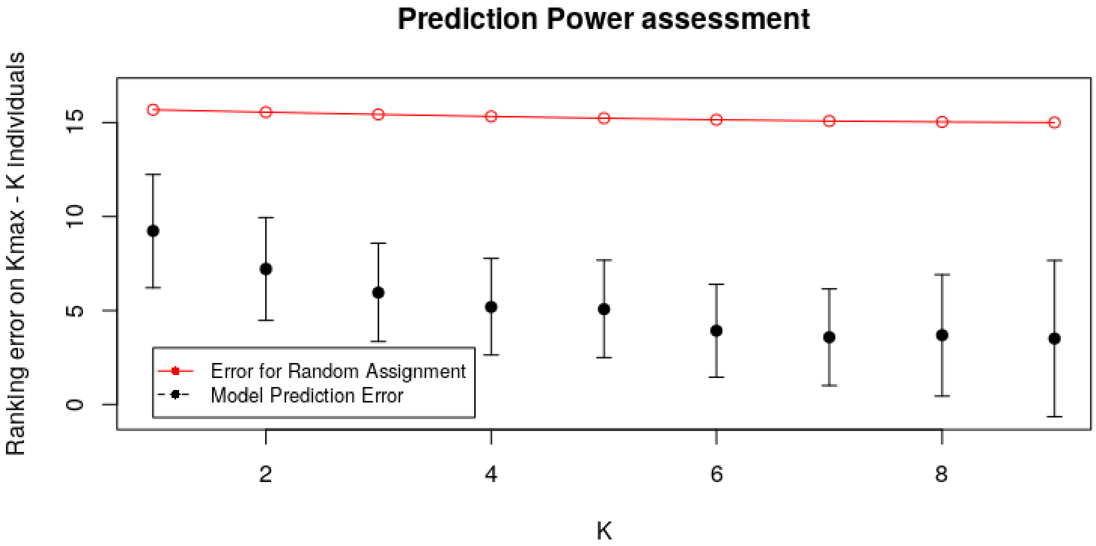


Figure 4.5: Assessment of the power of prediction of our model against a random permutation

We can see from the Figure 4.5, that the error from the model shrinks as K increases, which is what we expect since we predict the $K_{max} - K$ next preferred individuals with the information encapsulated in the K preferred individuals. Second, we can see that our model is efficient. With $K = 1$ we still predict 33% better than a random assignment for the prediction of the 9 next individuals in a network of 40 individuals. Similarly, the model performs 66% better than random for the prediction of the 10th individual, knowing the first 9.

4.1.2 Toy Example

In this subsection, we evaluate the coherence of the results obtained by our model and our algorithm by testing it on a toy model.

We drew a first simple example with a clear clustering into two groups of individuals, showed in Figure 4.6. Running the EM algorithm is instantaneous and returns the following matrix for W:

$$W = \begin{pmatrix} 0.1294021 & 1.0349799 \\ 0.1228317 & 0.8720837 \\ 0.1187870 & 0.7219490 \\ 1.0349782 & 0.1294015 \\ 0.8720818 & 0.1228311 \\ 0.7219470 & 0.1187863 \end{pmatrix} \tag{4.4}$$

There is a clear differentiation between two groups as illustrated in Figure 4.7. 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 bolder arrows), which is expected.

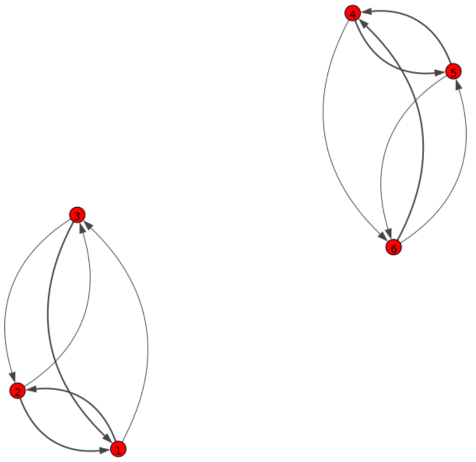


Figure 4.6: Initial Network with two independent groups of individuals

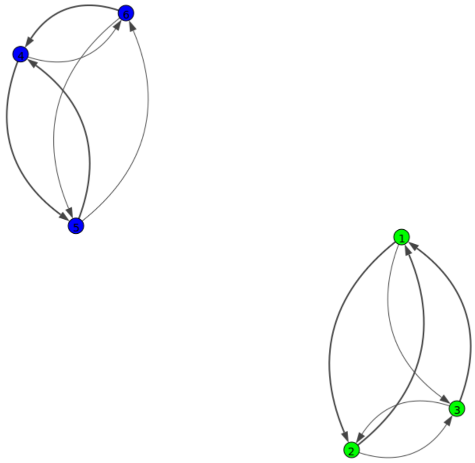


Figure 4.7: Inferred communities

Then, we drew a second example to test the recovering of missing data, showed in Figure 4.8.

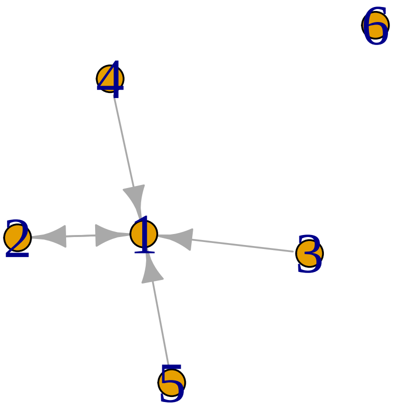


Figure 4.8: Toy network

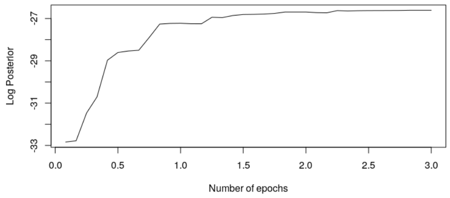


Figure 4.9: Posterior Inference

We obtain the vector

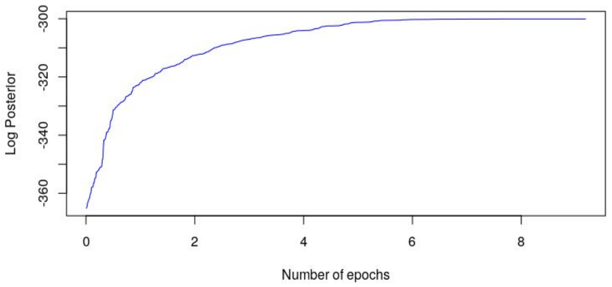
$$\lambda_6 = \begin{pmatrix} 0.7126756 \\ 0.3652081 \\ 0.2488320 \\ 0.2424432 \\ 0.2458105 \\ 0.0000000 \end{pmatrix} \tag{4.5}$$

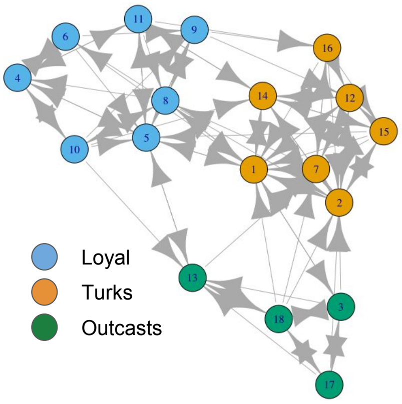
The model predicts that individual 6 is likely to prefer the most influential individual, individual 1, and then individual 2, which is expected.

4.1.3 Sampson Data Set

Here we test our model on a real-world data set, the Sampson data set, gathered by Samuel F. Sampson [2]. The data set summarizes relationships among 18 monks who were about to enter a monastery when a conflict erupted. The monks are divided by Sampson into three groups: Loyal Opposition, Turks, and Outcasts. The data set is in the form of rank preferences for each monk.

We ran the EM algorithm on this data set with a random initialization and $p = 3$ communities, and the same priors as previously used. We obtained the following result :





We recover the same communities as described by Sampson in his research.

We chose $p = 3$ because we already know from the study of Sampson that the data describe three factions. It can be interesting to infer the number of clusters directly from the data, thanks to the information criteria introduced previously. Here we computed both the AIC and the BIC of the data for different p, we computed the maximum likelihood estimators with a BFGS optimization algorithm [8]. We obtained plots 4.10 and 4.11:

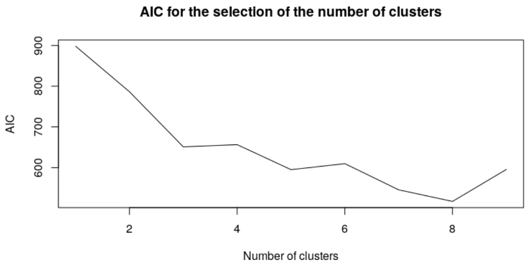


Figure 4.10: Model Selection with the AIC criterion

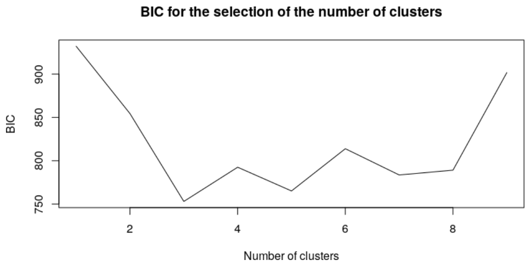


Figure 4.11: Model Selection with the BIC criterion

We can see from the plots that the AIC would select $p = 8$ clusters, whereas the BIC would select $p = 3$ clusters. This is expected since the BIC criterion discriminates complex models more than the AIC criterion. Although here the BIC criterion selects the expected model, these criteria are simple indications, model selection should be the result of a mature reflexion, and can take into account other elements such as time and memory costs.

4.2 Gibbs Sampler for Degree Correction

We implemented the Gibbs Sampler previously described in R, and made it able to support variable ranks and missing data, and hence any type of network as input. In the markov chain we only kept track of the $w_{i,j}$ and of the b_i to save memory, since the latent variables are useful for computational reasons. To perform inference, we chose to take the maximum a posteriori as point estimates. Indeed, as we will show in the plots of the posterior distribution, and discussed in the section Identifiability, a posterior mean estimate is not relevant since the priors and likelihood of our model are invariant by relabelling.

4.2.1 Synthetic Data

Auto-correlation

Here we plot the auto-correlation function for 10,000 samples drew from the gibbs sampler, with a random graph of size $n = 30$, with the following parameters : $K = 5, p = 4, a = 2, b = 2$. We averaged the auto-correlation over all the parameters in order to infer global information about the chain.

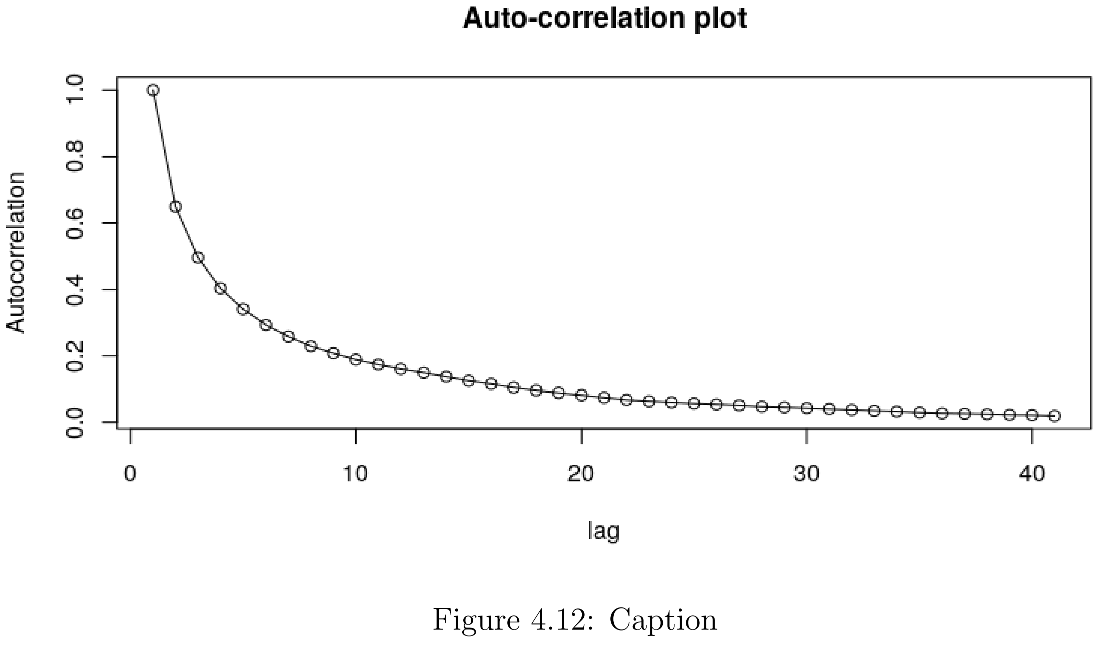


Figure 4.12: Caption

As expected, the auto-correlation tends to zero as the lag increases. We obtain for this simulation an effective sample size of $n_{ESS} = 13.35 \times n$. For lag = 10, the samples are weakly correlated. Hence, thinning the samples with a lag of 10 can optimize the use of memory, without losing too much information. A plot of the traces show that a small burn-in is sufficient to attain a state of "white noise" of the parameters.

4.2.2 Toy Example

Here we took the exact same example as in Figure 4.6, and ran the Gibbs sampler with $1,000,000$ samples with lag 10, and burn-in $20,000$. We obtain the following results:

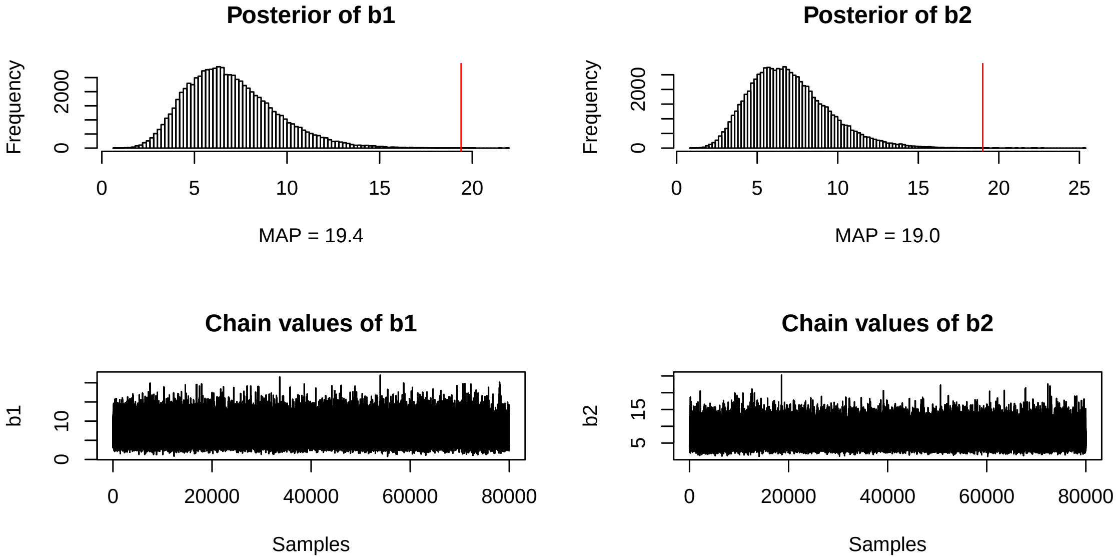


Figure 4.13: Gibbs Sampler for the Toy Example

$$Wmap = \begin{pmatrix} 0.078115599 & 0.0028624276 \\ 0.045019096 & 0.0003526772 \\ 0.069078810 & 0.0039184428 \\ 0.012006837 & 0.1371274807 \\ 0.025554788 & 0.1069460127 \\ 0.004471144 & 0.2250585478 \end{pmatrix} \tag{4.6}$$

We recover the same communities as before, the plots show that we obtain the same distribution for the b_i, which makes sense in this case. Taking the posterior mean estimates of the $w_{i,k}$ give the same communities, but the values obtained hardly differentiates the clusters (the w values are very similar).

4.2.3 Sampson Data Set

We ran $1,000,000$ iterations of the Gibbs Sampler, with $\alpha_0 = 1, \beta_0 = 1, a = 2, p = 3$ on the Sampson data set, and kept only one over ten samples in the chain. Fortunately, we recover exactly the same communities as before, as one can check on the resulting W written below, and we obtain the following point estimates for the degree corrections, based on the MAP of the Gibbs Sampler. The results are coherent since a high degree correction means that the community is less influent. Here the Outcasts are the least influent, then comes the Loyals, and finally the Turks. The Tuks and the Loyals have a similar degree of influence compared to the Outcasts.

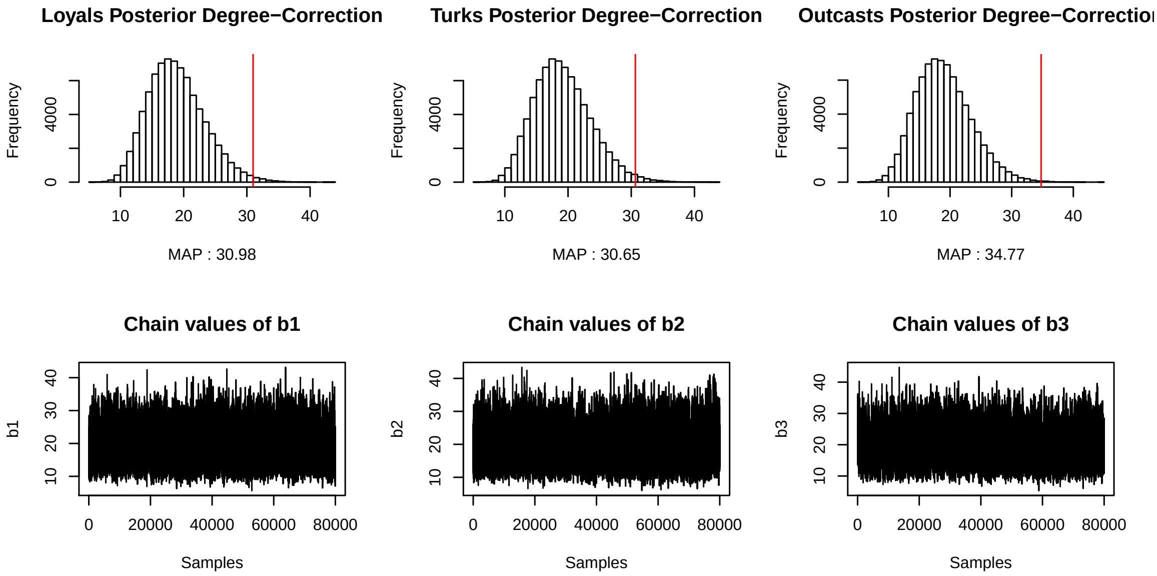


Figure 4.14: Gibbs Sampler for the Sampson Data Set

$$Wmap = \begin{pmatrix}
0.028342178 & 0.132794703 & 0.0028116997 \\
0.015526648 & 0.144659851 & 0.0225950927 \\
0.012170875 & 0.013111505 & 0.0435052950 \\
0.077882047 & 0.008449029 & 0.00663582510 \\
0.059784248 & 0.027460357 & 0.0158763848 \\
0.043725084 & 0.002394237 & 0.0017029809 \\
0.016711364 & 0.041887322 & 0.0258955378 \\
0.049441668 & 0.0131104534 & 0.0063907626 \\
0.084609098 & 0.008414093 & 0.0056879136 \\
0.011126239 & 0.001325130 & 0.0005210007 \\
0.032382804 & 0.014872686 & 0.0031258213 \\
0.001972192 & 0.041519433 & 0.0064190197 \\
0.007587735 & 0.004190892 & 0.0358233982 \\
0.005740627 & 0.067199814 & 0.0021474452 \\
0.008283047 & 0.034239040 & 0.0028699507 \\
0.004629673 & 0.028643459 & 0.0062985036 \\
0.003683552 & 0.003301192 & 0.0220477571 \\
0.001072851 & 0.012621787 & 0.0178437508
\end{pmatrix} \tag{4.7}$$

Again, taking the posterior mean estimates gives the same communities as the MAP, but the magnitude of differences between the $(w_{i,k})_{k\in[p]}$ is very small, due to the non-identifiability.

4.3 Model With Covariates

4.3.1 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 nodes, but liking only one individual. Here we have no covariates, only the parameters corresponding the popularity of each node.

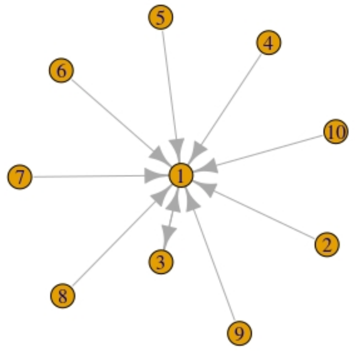


Figure 4.15: Graph Network Example

We obtain a popularity of $\beta_1 = 1.86$ for node 1, and $\beta_3 = 0.095$ for node 3 and $\beta^* = -0.28$ for the rest, which is consistent with what we would expect.

4.3.2 Peerinfl Data Set

The peerinfl data set was gathered by Daniel McFarland for its study on Student Resistance at school [15]. The data set consists of individual level attributes, such as how often does the student socialize or whether he likes the course on a scale of 0 to 5, as well as dyadic attributes, for the top 5 friends in semester one, and the whole class in semester two. We broke ties randomly for the top-5 friends and kept the classes that last the two semesters in order to incorporate the covariates, we obtained 25 classes on 36. Since certain students left after semester 1 and joined in semester 2, we only considered the one that stayed all along. Missing data were not frequent, hence we replaced them with the median value of the corresponding attributes. We deleted non-expected data, such as students belonging to their own top-5 friends. We also scaled the attributes by class.

Example

First we consider the class $n^o 851$, and the dyadic attributes consisting of the evaluation of i about j sociability, ability to work and expected grades.

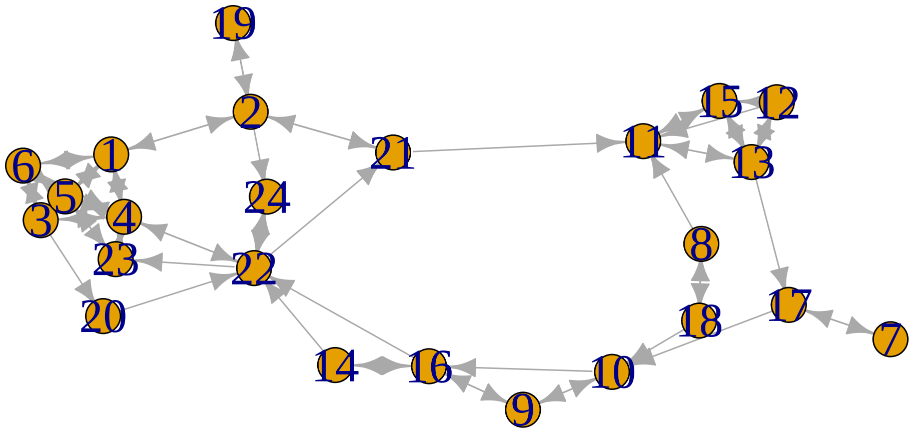


Figure 4.16: Class number 851 constituted of 24 students across two semesters.

We ran the Plackett-Luce model with covariates as well as the SRRM model with the package "amen" on R, [34], with $n = 100,000$ states and a burn-in of 1000. We obtained the following results :

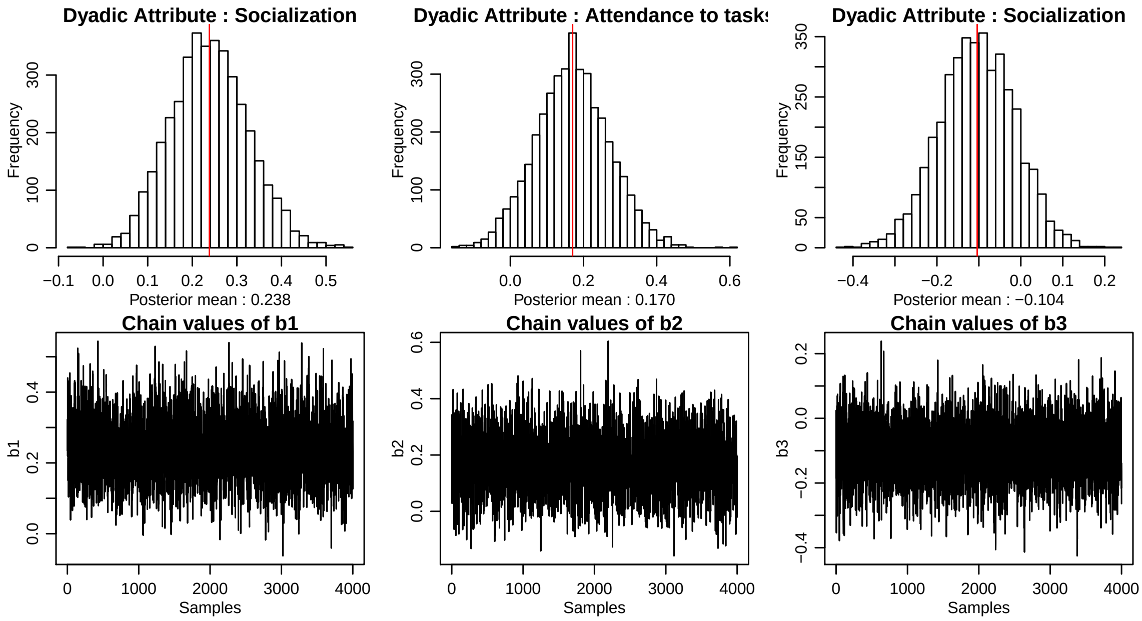


Figure 4.17: Estimation of the regression parameters of the SRRM model.

With Plackett-Luce, we respectively obtain $b_1 = 0.925, b_2 = 0.393, b_3 = 0.277$. The SRRM model allows us to compute the posterior mean estimates of the row and column effects of the model a_i and b_i, which are effectively correlated with the nodes respective Sociability and Popularity as we can see in the following plots.

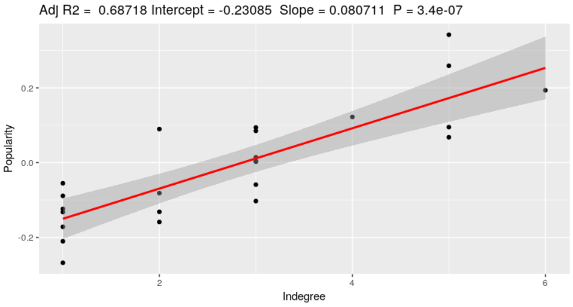


Figure 4.18: Correlation between b_i and the Indegree

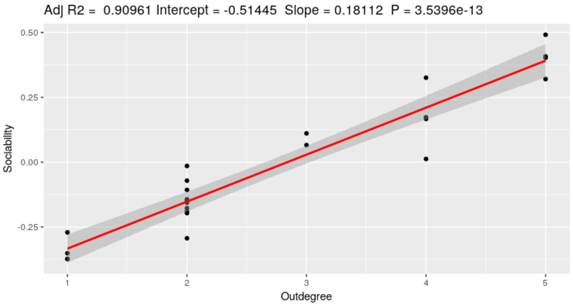


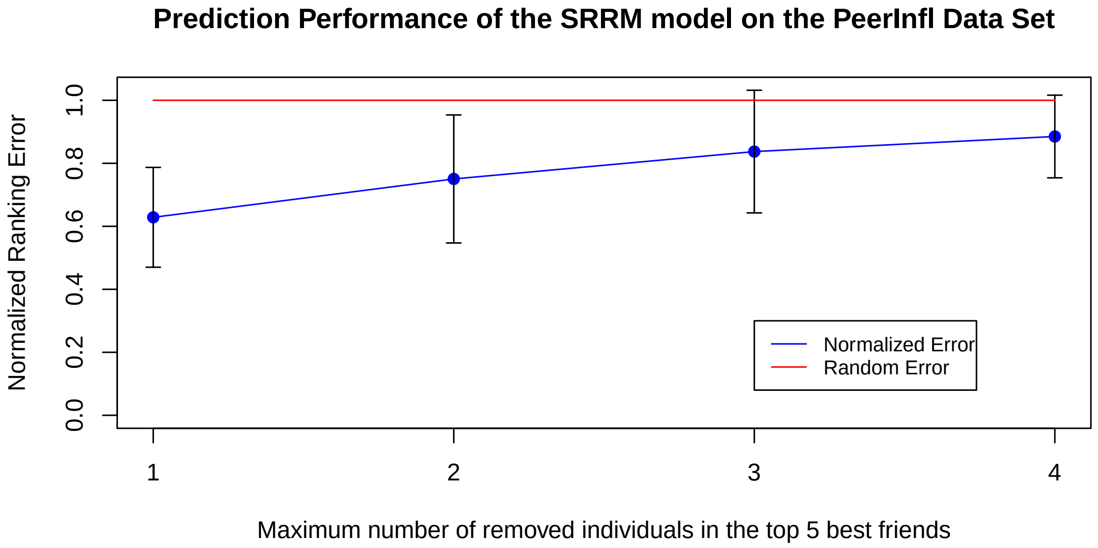
Figure 4.19: Correlation between a_i and the Outdegree

Prediction

Finally, we assess the prediction power of the SRRM model as follows :

1. For every class in the data set, we kept the students that stayed in both semesters, and filled missing data in the covariates with the median value of the attributes per class. We then built the network constituted by the maximum top-5 friends of each individuals in the class.
2. For every class network :
    (a) The nodes of the network do not have the same outdegree, we computed the maximum value of the outdegree K_{max}.
    (b) For $i \in [K_{max} - 1]$
        i. For every node k in the network, we removed the
        $min(i, outdegree(k) - 1)$ least important neighbours of k.
        ii. We fitted the SRRM model with the package "amen" from R, generating a Markov Chain of length $n_{scan} = 20,000$ with a burn-in of $1,000$.
        iii. We used the Y_{post} which is the posterior mean of the values of the matrix λ in our formulation to rank the other individuals of the network for each node k.
        iv. For every node k, we compared the predicted list with the actual best individuals that have been removed by computing the ratio between the ranking error of the individuals removed with the individuals predicted, and the random errors given by Eq.4.3. For every node k,
        $T = min(i, outdegree(k) - 1)$ and
        $n_{formula} = n_{network} - 1 - min(i, outdegree(k) - 1)$.
        v. We averaged the error over the nodes.
    (c) We stored the errors by i, the maximum number of individuals removed.
3. We averaged the errors over the classes by maximum number of individuals removed.

We obtain the following plot:



As we would expect, the error curve increases as we have less information to perform inference. For 1 and 2 individuals removed from the list of preferences, we are sure to recover the proper rankings better than randomly. For 3 and 4 individual removed, the model is better on average, but not in certain cases.

Chapter 5

Discussion

Many avenues remain to be investigated. First, the main drawback of the Plackett-Luce model as presented is the fact that the number of best friends chosen by an individual K_i is fixed in advance. Hence, the model cannot infer any information on the structure of the matrix λ given K_i. In particular, the "Sociability" or "Popularity" of an individual cannot be captured. This can be fixed by treating each K_i as a random variable depending on the λ, the scheme of variables can be interpreted in Figure 5.1. The distribution $P(K|\lambda)$ has to be designed to be analytically convenient and to represent some beliefs. Second, the Gibbs Sampler designed in Section "Degree-Correction" is correct but suffers the limitation that the posterior distribution of the parameters is invariant by relabelling, we have taken a MAP estimate to bypass this issue, but the downside of this estimate is that we cannot built confidence intervals contrary to the posterior mean estimate. Introducing artificial constraints on the prior could be advantageous as discussed in [14]. Third, a very interesting avenue of investigation would be to refine our models with covariates to perform community detection. This could be done by introducing a mixture of Plackett-Luce models as in [26]. One could explore its adaption to the SRRM model. Fourth, many realistic data have ties in their rank nominations, in the case pairwise comparison, some model

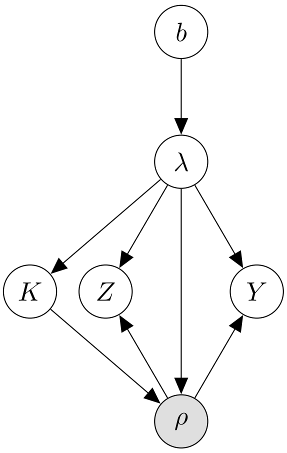


Figure 5.1: Adaptation of the Plackett-Luce model for partial rankings

treating ties in the rankings have been developed [3], [27], one could investigate their adaptation in the case of the listwise approach. Finally, deep learning approaches have been developed to model λ as a function of covariates with a neural net [20], the parameters being trained with a loss function and an optimization algorithm (typically by gradient descent), one could investigate the virtues of these approaches with respect to the methods developed in this project.

Appendices

Appendix A

Preliminaries

A.1 MM algorithm

The MM algorithm (Majorize-Minimization or Minorize-Maximization) is a procedure for building an iterative optimization algorithm [18]. The MM algorithm works by finding a surrogate function that minorizes or majorizes the objective function. Optimizing the surrogate function will drive the objective function upward or downward until a local optimum is reached. In the case of a concave $f(\theta)$ function to be maximized the algorithm works as follow, at step m (Minorize-Maximization):

1. Build a surrogate function $g(\theta|\theta_m)$ such that:
    (a) $g(\theta|\theta_m) \geq f(\theta)\ \forall\theta$
    (b) $g(\theta_m|\theta_m) = f(\theta_m)$
2. Maximize $g(\theta|\theta_m)$ instead of $f(\theta)$ and let
$$\theta_{m+1} = \operatorname*{argmax}_{\theta} g(\theta|\theta_m)$$
3. Iterate until converge.

The Maximization-Minorize is similar for a convex objective function.

A.2 EM algorithm

The expectation maximization algorithm, or EM algorithm, is a general technique for finding maximum likelihood solutions for probabilistic models having latent variables [5], [13] and can be treated as a special case of the MM algorithm.

If we consider a model with X the observed variables and Z the latent variables, and θ the set of parameters that govern the likelihood of the model, we want to maximize the quantity:

$$p(X|\theta) = \int_Z p(X, Z|\theta) dZ \tag{A.1}$$

In the case where this integral is intractable, the EM algorithm is an applicable alternative that finds the MLE by iterative updates. It proceeds in two steps:

E-step(Expectation Step): Define $Q(\theta|\theta^{(t)})$ as the expected value of the log likelihood function of θ, with respect to the current conditional distribution of Z given X and the current estimates of the parameters $\theta^{(t)}$

$$Q(\theta|\theta^{(t)}) = \mathbb{E}_{Z|X,\theta^{(t)}}[log(p(X,Z|\theta))] \tag{A.2}$$

M-step(Maximization Step): Find the parameters that maximize this quantity:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}) \tag{A.3}$$

The algorithm then iterates these two steps until convergence. This algorithm monotonically approaches a local minimum of the cost function.

A.3 Gibbs Sampler

The Gibbs Sampler is a particular case of a Markov Chain Monte Carlo algorithm. The goal is to sample from a target distribution with multiple variables $\pi(x_1, ..., x_n)$. In the configuration of the Gibbs sampler, we can sample from the conditional distribution $\pi(x_i|(x_j)_{j\neq i})$, we then have the following steps [10]:

1. We begin with initial values $X^{(t)} = (x_1^{(t)}, ..., x_n^{(t)})$.
2. Then, to build the full vector $X^{(t+1)}$, we sample coordinate-wise according to the conditional distributions:

$$x_i^{(t+1)} \sim \pi(x_i|x_1^{(t+1)}, ..., x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, ..., x_n^{(t)}) \tag{A.4}$$

3. We repeat the previous step k times, for a chain of length $k + 1$.

The samples generated by this process approximate the joint distribution $\pi(x_1, ..., x_n)$. From the samples we can approximate the marginal distribution over a subset of the variables by just observing this subset and ignoring the rest. The average of a certain variable over the samples is also an approximation of the mean of the variable. A burn-in step in which a predetermined number of first samples are removed from the chain.

A.4 Metropolis-Hasting Algorithm

The Metropolis-Hasting Algorithm is a particular case of a Markov Chain Monte Carlo algorithm. The goal is to sample from a target distribution $\pi(x)$. In the configuration of the Metropolis-Hasting Algorithm, we can evaluate a function $f(x)$ proportional to the target distribution $\pi(x)$, we, then, have the following steps [19]:

1. Initialization: We choose a initial sample x_0 and a proposal distribution $Q(x|y)$ that proposes the next candidate for the Markov Chain, given a previous state.

2. For each iteration t:

   (a) **Generate** : Generate a candidate x' for the next sample by picking from the distribution $Q(x'|x^{(t)})$

   (b) **Calculate** : Calculate the acceptance ratio $\alpha = min(1, \frac{f(x')Q(x^{(t)}|x')}{f(x^{(t)})Q(x'|x^{(t)})})$, which will be used to decide whether to accept or reject the candidate.

   (c) **Accept or Reject** : Sample $u \sim \mathcal{U}(0,1)$. If $u < \alpha$ accept the proposition and assign $x^{(t+1)} = x'$, else reject and assign $x^{(t+1)} = x^{(t)}$.

In this algorithm, a burn-in step in which a predetermined number of first samples are retrieved from the chain can be applied in order to reduce the effect of the first samples that can follow a completely different distribution. Furthermore, the samples generated by this algorithm are typically auto-correlated. Thus, we should throw away the majority of the samples and only take every *nth* sample, for some value of n.

A.5 Reversible Jump MCMC

The Reversible Jump MCMC (RJMCMC) is a Markov Chain Monte Carlo method that supports jump across dimensions, for the purpose of model selection when the number of parameters is not the same in the models. The chain targets the posterior density $p(\theta, m|D)$. Let $(m^{(t)}, \theta^{(t)}_{m^{(t)}})$ be the current state of the Markov Chain, the next iteration is constructed as follows (see [30]):

1. Sample a candidate model $M^*|m^{(t)}$ from a proposal density with conditional density $g(|m^{(t)})$.

2. Given $M^* = m^*$ generate an augmenting variable $U|(m^{(t)}, \theta^{(t)}_{m^{(t)}}, m^*)$ from a proposal distribution with density $h(.|m^{(t)}, \theta^{(t)}_{m^{(t)}}, m^*)$. Let

$$(\theta^{(t)}_{m^{(t)}}, U^*) = q_{t,*}(\theta^{(t)}_{m^{(t)}}, U) \tag{A.5}$$

   Where $q_{t,*}$ is an invertible mapping from $(\theta^{(t)}_{m^{(t)}}, U)$ to $\theta^*_{m^*}, U^*$ and the auxiliary variables have dimensions satisfying $dim(\theta^{(t)}_{m^{(t)}}) + dim(U) = dim(\theta^*_{m^*}) + dim(U^*)$.

3. For a proposed model, $M^* = m^*$, and the corresponding proposed parameter values $\theta^*_{m^*}$, compute the Metropolis - Hastings acceptance probability given by:

$$\alpha(x^t, x^*) = min(1, \frac{p(m^*, \theta^*_{m^*}, U^*|y)g(m^{(t)}|m^*)h(u^*|m^*, \theta^*_{m^*}, m^{(t)})}{p(m^{(t)}, \theta^{(t)}_{m^{(t)}}, U^{(t)}|y)g(m^*|m^{(t)})h(u|m^{(t)}, \theta^{(t)}_{m^{(t)}}, m^*)}|J(t)|) \tag{A.6}$$

Where where J(t) is the Jacobian matrix:

$$J(t) = \frac{dq_{t,*}f(\theta,u)}{d(\theta,u)}\Big|_{(\theta,u)=(\theta^{(t)}_{m^{(t)}},U)} \tag{A.7}$$

If the proposal is accepted, set $x^{(t+1)} = (m^*, \theta^*_{m^*})$, otherwise set $x^{(t+1)} = x^{(t)}$.

4. Discard U and U^* and return to step 1.

A.6 MCMC Convergence Diagnostic

When inferring parameters using MCMC methods, the question of convergence and accuracy is crucial, since we study a probability distribution with finite Markov Chains that are supposed to converge asymptotically. Convergence diagnostics is a complex subject and well studied by the literature [21]. We will only describe some tools.

1. When studying a Markov Chain, the first tool we can use is the *trace plot*, which is simply the plot of the parameters of the markov chain. Generally speaking, the trace plot consists of a "transitional regime" converging toward a "permanent regime" where the parameters behave like random noise around a mean value. In this case, the transitional regime can be considered as irrelevant, and be dropped in the Markov Chain in order to increase the accuracy of some estimators. The number of initial states that we may drop is called the *Burn in*.

2. The main objective of a Markov Chain in a Bayesian configuration is to sample from the posterior distribution of the parameters. Since every new state of a Markov Chain is constructed with the previous one, the samples are typically correlated. This correlation depends on the way the Markov Chain is designed. For instance, in a M-H algorithm, a Markov Chain with a very low acceptance ratio results in many constant states, hence many fully correlated states. On the contrary, when the acceptance ratio is too high, the proposal may be too concentrated on the previous state in the space of parameters, which also mean that the states are correlated. This correlation results in a loss of information compared to totally independent states, and thus a waste of memory. To assess the correlation between states, the *Autocorrelation* function can be used and is defined, for lag s, and parameter random variable X_i from the i_{th} state of the chain by:

$$\rho_s = \frac{Cov(X_i, X_{i+s})}{Var(X_i)} \tag{A.8}$$

This quantity can be estimated on the whole chain. When one plot the autocorrelation as a function of the lag s, a fast rate of decay of the curve obtained indicates that the chain is weakly correlated, which is desirable. To optimize the memory cost, one can design a lag parameter for the markov chain, which consists in preserving the states of the chain only every s^{th} moves. The states dropped are informative, but their removal optimizes memory cost. To assess the impact of state correlation on the efficiency of the Markov Chain, one can compute the effective sample size n_{ess} which is the equivalent length of a chain of independant states:

$$n_{ess} = \frac{n}{\tau} \tag{A.9}$$

$$\tau = 1 + 2\sum_{s=1}^{n-1} \rho_s \tag{A.10}$$

Often, τ is estimated up to $t < n - 1$ since $\lim\limits_{s \to +\infty} \rho_s = 0$.

Bibliography

[1] RD Luce. *Individual choice theory: A theoretical analysis.* 1959.

[2] Samuel F Sampson. “A novitiate in a period of change: An experimental and case study of social relationships.” In: (1969).

[3] Roger R Davidson. “On extending the Bradley-Terry model to accommodate ties in paired comparison experiments”. In: *Journal of the American Statistical Association* 65.329 (1970), pp. 317–328.

[4] Robin L Plackett. “The analysis of permutations”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24.2 (1975), pp. 193–202.

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[6] Rebecca M Warner, David A Kenny, and Michael Stoto. “A new round robin analysis of variance for social interaction data.” In: *Journal of Personality and Social Psychology* 37.10 (1979), p. 1742.

[7] Randall G Chapaaan and Richard Staelin. “Exploiting rank ordered choice set data within the stochastic utility model”. In: *Journal of marketing research* 19.3 (1982), pp. 288–301.

[8] Roger Fletcher. “Practical methods of optimization john wiley & sons”. In: *New York* 80 (1987).

[9] Dankmar Böhning and Bruce G Lindsay. “Monotonicity of quadratic-approximation algorithms”. In: *Annals of the Institute of Statistical Mathematics* 40.4 (1988), pp. 641–663.

[10] George Casella and Edward I George. “Explaining the Gibbs sampler”. In: *The American Statistician* 46.3 (1992), pp. 167–174.

[11] Hirotugu Akaike. “Implications of informational point of view on the development of statistical science”. In: *Selected Papers of Hirotugu Akaike.* Springer, 1994, pp. 421–432.

[12] Peter J Green. “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination”. In: *Biometrika* 82.4 (1995), pp. 711–732.

[13] GJ McLachlan and T Krishnan. “The EM algorithm and extensions John Wiley & Sons”. In: *Inc., New York* (1997).

[14] Sylvia Richardson and Peter J Green. “On Bayesian analysis of mixtures with an unknown number of components (with discussion)”. In: *Journal of the Royal Statistical Society: series B (statistical methodology)* 59.4 (1997), pp. 731–792.

[15] Daniel A McFarland. “Student resistance: How the formal and informal organization of classrooms facilitate everyday forms of student defiance”. In: *American journal of Sociology* 107.3 (2001), pp. 612–678.

[16] Rasmus Waagepetersen and Daniel Sorensen. “A Tutorial on Reversible Jump MCMC with a View toward Applications in QTL-mapping”. In: *International Statistical Review* 69.1 (2001), pp. 49–61.

[17] David R Hunter et al. “MM algorithms for generalized Bradley-Terry models”. In: *The annals of statistics* 32.1 (2004), pp. 384–406.

[18] David R Hunter and Kenneth Lange. “A tutorial on MM algorithms”. In: *The American Statistician* 58.1 (2004), pp. 30–37.

[19] Bernd A Berg and Alain Billoire. “Markov chain monte carlo simulations”. In: *Wiley Encyclopedia of Computer Science and Engineering* (2007).

[20] Zhe Cao et al. “Learning to rank: from pairwise approach to listwise approach”. In: *Proceedings of the 24th international conference on Machine learning*. ACM. 2007, pp. 129–136.

[21] Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[22] Isobel Claire Gormley, Thomas Brendan Murphy, et al. “A mixture of experts model for rank data with applications in election studies”. In: *The Annals of Applied Statistics* 2.4 (2008), pp. 1452–1477.

[23] John Guiver and Edward Snelson. “Bayesian inference for Plackett-Luce ranking models”. In: *proceedings of the 26th annual international conference on machine learning*. ACM. 2009, pp. 377–384.

[24] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix factorization techniques for recommender systems”. In: *Computer* 8 (2009), pp. 30–37.

[25] Harish S Bhat and Nitesh Kumar. “On the derivation of the Bayesian Information Criterion”. In: *School of Natural Sciences, University of California* (2010).

[26] Isobel Claire Gormley and Thomas Brendan Murphy. “Clustering ranked preference data using sociodemographic covariates”. In: *Choice Modelling: The State-of-the-art and The State-of-practice: Proceedings from the Inaugural International Choice Modelling Conference*. Emerald Group Publishing Limited. 2010, pp. 543–569.

[27] Francois Caron and Arnaud Doucet. “Efficient Bayesian inference for generalized Bradley–Terry models”. In: *Journal of Computational and Graphical Statistics* 21.1 (2012), pp. 174–196.

[28] David R Hunter, Pavel N Krivitsky, and Michael Schweinberger. “Computational statistical methods for social network models”. In: *Journal of Computational and Graphical Statistics* 21.4 (2012), pp. 856–882.

[29] Elchanan Mossel, Joe Neeman, and Allan Sly. “Stochastic block models and reconstruction”. In: *arXiv preprint arXiv:1202.1499* (2012).

[30] Paolo Giudici, Geof H Givens, and Bani K Mallick. *Wiley Series in Computational Statistics*. Wiley Online Library, 2013.

[31] Peter Hoff et al. “Likelihoods for fixed rank nomination networks”. In: *Network Science* 1.3 (2013), pp. 253–277.

[32] Mark EJ Newman. “Spectral methods for community detection and graph partitioning”. In: *Physical Review E* 88.4 (2013), p. 042822.

[33] Charles Bordenave, Marc Lelarge, and Laurent Massoulié. “Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs”. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 1347–1357.

[34] Peter D Hoff. “Dyadic data analysis with amen”. In: *arXiv preprint arXiv:1506.08237* (2015).

[35] Pavel N Krivitsky and Carter T Butts. “Exponential-family random graph models for rank-order relational data”. In: *Sociological Methodology* 47.1 (2017), pp. 68–112.