Modeling and Analysis of Social Network Data from Rank Preferences

1032067

Dissertation for
Master of Science in Mathematical Sciences (OMMS)
Trinity 2019

Contents

Abstract

Given a group of individuals and their ranked preferences of other individuals in the group, a social network can be constructed such that a directed edge from individual i to j indicates i named j among their top connections. With this data of ranked preferences, we are interested in estimating the "affiliation levels" among the individuals in the network. We introduce a latent community representation for these affiliation levels that allows for dimensionality reduction and soft clustering. A Bayesian approach with the popular Plackett-Luce model is used to analyze the ranking data, and we derive Markov Chain Monte Carlo (MCMC) and Expectation Maximization (EM) algorithms for parameter estimation. The goal is then twofold: to predict or recommend further preferences of individuals and to cluster individuals into distinct communities. The derived model, particularly the EM algorithm, is shown to be flexible in accurately addressing both of these questions.

1 Introduction

Social networks is a varied and widely researched sub-field of network analysis. As technology enables society to become more connected, and as networks grow from relatively small collections of local individuals to thousands of connected individuals across the globe, we become increasingly interested in analyzing them to draw insights about their nodes and structure. More available information leads to some core questions about the status and relationships between nodes, the tendencies for certain nodes to connect based on some shared characteristics, and the underlying communities or sub-communities in a network. This analysis focuses on the strength of relationships between individuals in a network and what information we could draw from them.

With this in mind, given a network of individuals and their lists of ranked preferences, we aim to devise a method to estimate the "affiliation level" between each of the individuals. Naturally, a higher affiliation level indicates a stronger relationship between two individuals. In this context, the ranked preference list of an individual is a list of their top connections in the network. We can imagine, for example, that this data is collected by surveying a subset of the individuals and asking them to list up to their top five closest friends in the network. We denote these ranked preference lists as $\rho_i = (\rho_{i1}, ..., \rho_{ik_i})$

to mean the top k_i list of friends given by individual i, where ρ_{i1} is their best friend, ρ_{i2} is their second best friend and so on up to k_i. Note that we use k_i to accommodate different size lists. Once these affiliation levels have been estimated, they could allow for performing additional tasks like soft clustering of the individuals into communities and predicting further connections or connections of individuals who have not provided a list.

Ranking data collected this way under what is called a fixed rank nomination (FRN) scheme is a common surveying technique for social networks. It provides relations up to a certain fixed number but leaves the existence of any further relations unknown [6]. One example of such data is the National Longitudinal Study of Adolescent to Adult Health (Add Health) to examine adolescent high school relationships [9]. Another older but smaller example which has been a staple in social network analysis is Sampson's Monastery data that records the relations among monks entering a monastery [14]. Previous studies have looked at using this ranking data format to analyze social networks. For instance, Fosdick, Hoff, Volfovsky and Stovel developed a set-based likelihood and simulation method specifically for FRN data. They analyzed the Add Health data to perform inference on the parameters dictating a social relations regression model to ultimately estimate what effect varying individual characteristics, activities and behaviors have on students who evaluate other students as friends [6].

We will focus on Sampson's Monastery data and estimate the aforementioned affiliation levels. To do so, we use the Plackett-Luce model [12, 13], a well-known model that looks at the distribution over the permutations of objects in a list. In this context, the objects are individuals in the network.

Using Plackett-Luce to model ranking data is well-studied. Previous applications include placings in NASCAR races [3, 8, 10] and voting via candidate preferences in Irish elections [7]. In both cases, the set of objects i providing the rankings and the set of objects being ranked are independent. More specifically, for the NASCAR example, i is essentially an index representing the race number and the objects being ranked are the race cars. The goal there is to estimate the skill levels of each of the racers. Predictions for the rankings are influenced by rankings in other races, but not by the actual race number. Similarly in the Irish voting study, constituent i ranks the candidates, but the order of the candidates is not influenced by any attributes of the constituent. The interest in applying the Plackett-Luce model to a social network is that it introduces an additional consideration, where the set of individuals providing the ranked lists and the set of individuals being

ranked are the same. Hence, individual j being ranked by individual i is partially influenced by whether or not i appears on j's list. Additionally, the recent rise in social media and extremely large social networks present new potential applications. Though we work with a relatively small data set here, in a broad sense using ranking data to gauge relations in a network could be applied to large-scale problems in social media such as recommending new friends and connections based on the interactions among already existing connections.

The rest of this section gives a preliminary overview of the tools we will use for parameter estimation, namely Expectation-Maximization (EM) and Markov Chain Monte Carlo (MCMC) algorithms. Section 2 introduces the simple Bradley-Terry model, the extension to Plackett-Luce, and the usage of latent variables to derive the algorithms. We simulate data and test the model in section 3. We apply the model to the Sampson's monastery data in section 4, and a brief discussion follows in section 5.

1.1 Preliminaries

We give a brief overview of EM and MCMC as these are the two algorithms we will derive. For the EM algorithm, it is necessary to introduce unobserved latent data Z such that the likelihood function is $\mathbb{P}(X, Z|\theta)$ where X is the observed data and θ are the unknown parameters. The maximum likelihood estimate (MLE) is found from maximizing the marginal likelihood of X, given as:

$$L(\theta; X) = \mathbb{P}(X|\theta) = \int_Z \mathbb{P}(X, Z|\theta) dZ \tag{1.1}$$

which is intractable. The EM algorithm then proceeds iteratively in two steps. Given the observed data X and the current parameter estimates θ^*, the expectation step is defined as the expected value of the *complete data* log likelihood with respect to the conditional distribution of the latent variable. We denote this as:

$$Q(\theta, \theta^*) = \mathbb{E}_{Z|X,\theta}[\ell_c(\theta; X, Z)] \tag{1.2}$$

For the maximization step, we set the derivative to 0 and solve for θ to get the new set of estimates θ^{**}:

$$\theta^{**} = \underset{\theta}{\operatorname{argmax}}\ Q(\theta, \theta^*) \tag{1.3}$$

We iterate this until the parameter estimates do not change beyond a certain threshold to obtain a local MLE. Note that adding a prior distribution to the expectation step and running the algorithm will return a local maximum a posteriori estimate (MAP). Since the estimates are local and not global, the algorithm is run with different initial values of θ to check stability. With each iteration, the EM algorithm increases the observed likelihood [4].

Where EM is deterministic and returns a point estimate of the MLE or MAP, MCMC is random and returns samples from a target distribution. Computing the mean from this distribution will give the desired parameter estimate. Broadly speaking, MCMC methods are used to sample from a complex distribution where direct sampling is not possible. This is often a posterior distribution to find the MAP. The idea is to construct a Markov chain with a stationary distribution proportional to the target distribution so that after some number of steps, called the burn-in period, we are effectively sampling from the target distribution. We will derive a Gibbs sampling scheme which requires sampling from the conditional distributions of the target distribution. This follows naturally from the latent variables and EM setup, similarly to how it is derived by Caron and Doucet [3]. Suppose we have observed X, latent Z_i, and parameters θ_i and want to sample from the posterior distribution $\mathbb{P}(\theta, Z|X)$. This can be done at iteration t with the following sampling method:

For $i = 1, ..., n$:

$$Z_i^{(t)} \sim \mathbb{P}(Z_i^{(t)}|X, \theta^{(t-1)}) \tag{1.4}$$

$$\theta_i^{(t)} \sim \mathbb{P}(\theta_i^{(t)}|X, Z^{(t)}) \tag{1.5}$$

As we are using Markov chains where each sample depends only on the previous values, we check autocorrelation and traceplots to see if the chain mixes well. If after a certain number of iterations autocorrelation is low and tends to 0 and traceplots show no signs of increasing or decreasing patterns, we can conclude the samples are effectively independent.

2 Plackett-Luce Model

To understand the Placket-Luce model [12, 13], we introduce the basic Bradley-Terry model, first proposed by Bradley and Terry in 1952 [2]. Given a set of elements that are repeatedly compared in pairs, elements i and j are compared with the probability that i beats j as:

$$\mathbb{P}(i > j) = \frac{\lambda_i}{\lambda_i + \lambda_j} \tag{2.1}$$

where λ_i is a parameter representing the "skill level" of element i. The standard Plackett-Luce model extends Bradley-Terry to consider comparisons between more than two elements. Suppose k_i elements are ranked for comparisons indexed by $i = 1, ...n$. Let $\rho_i = (\rho_{i1}, ..., \rho_{ik_i})$ be the ranking of the elements up to k_i for comparison i. Note that this notation is identical to what is defined in section 1, but the meaning is slightly different. The standard Plackett-Luce model for $i = 1, ..., n$ and $m = 1, ..., k_i$ is:

$$\mathbb{P}(\rho_i|\lambda) = \prod_{m=1}^{k_i-1} \frac{\lambda_{\rho_{im}}}{\sum_{j=m}^{k_i} \lambda_{\rho_{ij}}} \tag{2.2}$$

where λ still represents skill level. Similarly to Gromley and Murphy's study [7], the first factor in (2.2) can be understood as the probability that element ρ_{i1} is ranked first in comparison i.

To adapt this model to a social network, we first reinterpret ρ_i as it is defined in section 1 to be the top k_i individuals listed by individual i. We define λ_{ij} to be the affiliation level between individuals i and j. We also note n is both the number of comparisons and the number of individuals in the network. The modified Plackett-Luce model is:

$$\mathbb{P}(\rho_i|\lambda) = \frac{\lambda_{i\rho_{i1}}}{\sum_{j\neq i} \lambda_{ij}} \times ... \times \frac{\lambda_{i\rho_{ik_i}}}{\sum_{j\neq i} \lambda_{ij} - \sum_{l=1}^{k_i-1} \lambda_{i\rho_{il}}} = \prod_{m=1}^{k_i} \frac{\lambda_{i\rho_{im}}}{\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}} \lambda_{ij}} \tag{2.3}$$

where $\rho_{<im} := \{\rho_{i1}, ..., \rho_{im-1}\}$ is the set of individuals ranked higher than m by individual i. For each ranked list i, regardless of what k_i is, every other individual in the network should be considered. In other words, unless $k_i = n - 1$, we are looking at partial rankings. This is reflected in the denominator of (2.3), where the first factor contains every other element in

the network not equal to i. Note that this is different from (2.2), which is set up so each comparison i only considers the k_i *participating* elements in that comparison. Only the parameter values associated with the elements to be ranked for i will appear in the product. Essentially, each comparison models the full rankings *within* that comparison.

2.1 Direct Estimation on λ

We first derive a model for direct estimation on λ. Bradley-Terry models and their extensions like Plackett-Luce can take on the following Thurstonian interpretation [5, Chapter 9]: for each affiliation between i and j, let latent variables $Y_{ij} \sim \varepsilon(\lambda_{ij})$ where ε is the exponential distribution and λ_{ij} is the rate parameter. We interpret these Y_{ij} as arrival times in a Poisson process so that a higher affiliation - or rate parameter - corresponds to a faster arrival time. The individual with the fastest arrival time according to i is ranked first by i. We then have:

$$Y_{i\rho_{i1}} < Y_{i\rho_{i2}} < ... < Y_{i\rho_{ik_i}}$$

This is the order of the top k_i preferences, or connections, of individual i. Note that ρ_{ik_i} is simply some other individual j. Rather than use these latent variables to derive the EM algorithm, we follow Caron and Doucet [3] and introduce another latent random variable Z such that $Z_{i1} = \min(Y_{ij})$. By properties of exponential distributions, $Z_{i1} \sim \varepsilon(\sum_{j\neq i} \lambda_{ij})$. This denotes the distribution of arrival times of the first preference of individual i. Similarly, $Z_{i2} = \min(Y_{ij} \text{ excluding } Y_{i\rho_{i1}}) \sim \varepsilon(\sum_{j\neq i, j\neq \rho_{i1}} \lambda_{ij})$. Generalizing for the m-th preference:

$$Z_{im} = \min(Y_{ij} \text{ excluding } Y_{i\rho_{i1}} : Y_{i\rho_{im-1}}) \sim \varepsilon(\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}} \lambda_{ij})$$

with $\rho_{<im}$ defined from (2.3). We use latent Z since it is a deterministic function of Y and will have less missing information, which will lead to faster convergence of the EM and MCMC algorithms [11, Chapter 6]. The likelihood of latent variable Z is:

$$p(z|\rho,\lambda) = \prod_{i=1}^{n}\prod_{m=1}^{k_i} \varepsilon(z_{im}; \sum_{\substack{j\neq i \\ j\notin\rho_{<im}}} \lambda_{ij}) \tag{2.4}$$

The likelihood of the Plackett-Luce model is:

$$p(\lambda;\rho) = \prod_{i=1}^{n}\prod_{m=1}^{k_i} \frac{\lambda_{i\rho_{im}}}{\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}} \lambda_{ij}} \tag{2.5}$$

Multiplying (2.4) and (2.5) to get the complete likelihood, canceling out the denominator of the Plackett-Luce likelihood with the rate parameter leading coefficient in the exponential distribution of Z, and taking the log gives the complete data log likelihood:

$$\ell_c(\lambda) = \sum_{i=1}^{n}\sum_{m=1}^{k_i}\left[\log\lambda_{i\rho_{im}} - z_{im}\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}}\lambda_{ij}\right] \tag{2.6}$$

To maximize the log-posterior with the EM algorithm, we follow Guiver and Snelson [8] and assign a Γ prior to λ with the following likelihood:

$$p(\lambda) = \prod_{i=1}^{n}\prod_{j=1}^{n}\Gamma(\lambda_{ij};a,b) \tag{2.7}$$

The $Q(\lambda,\lambda^*)$ function is:

$$\begin{aligned} Q(\lambda,\lambda^*) &= \mathbb{E}_{Z|\rho,\lambda^*}[\ell_c(\lambda)] + \log p(\lambda) \\ &\equiv \sum_{i=1}^{n}\sum_{m=1}^{k_i}\left[\log\lambda_{i\rho_{im}} - \frac{\sum_{j\neq i, j\notin\rho_{<im}}\lambda_{ij}}{\sum_{j\neq i, j\notin\rho_{<im}}\lambda^*_{ij}}\right] + \sum_{i=1}^{n}\sum_{j=1}^{n}\left[(a-1)\log\lambda_{ij} - b\lambda_{ij}\right] \end{aligned}$$

where "$\equiv$" is "equal but excluding terms independent from the first argument of $Q(\lambda,\lambda^*)$". In this case, b^a and $\Gamma(a)$ from the Γ distributed prior are ignored. Finding the argmax for λ_{ij} gives the EM update step at time t:

$$\lambda_{ij}^{(t)} = (a-1+n_{ij})\left[b + \sum_{m=1}^{k_i}\left(\frac{\delta_{ijm}}{\sum_{j\neq i, j\notin\rho_{<im}}\lambda_{ij}^{(t-1)}}\right)\right]^{-1} \tag{2.8}$$

where:

$$n_{ij} = \begin{cases} 1 \text{ if } j\in\rho_i \\ 0, \text{ otherwise} \end{cases}$$

and:

$$\delta_{ijm} = \begin{cases} 1 \text{ if } j \notin \rho_{<im} \\ 0, \text{ otherwise} \end{cases}$$

so n_{ij} indicates whenever j is listed by i, and δ_{ijm} simply indicates whenever j has not been ranked higher than m by i. This form and notation is similar to that of Hunter [10]. We can sample from the posterior $p(z, \lambda|\rho)$ with the following MCMC algorithm at time t:

For $i = 1, ..., n$ and $m = 1, ..., k_i$:

$$Z_{im}^{(t)}|\rho, \lambda^{(t-1)} \sim \varepsilon \left(\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \lambda_{ij}^{(t-1)} \right) \tag{2.9}$$

For $i, j = 1, ...n$ and $i \neq j$:

$$\lambda_{ij}^{(t)}|\rho, Z^{(t)} \sim \Gamma \left(a + n_{ij},\ b + \sum_{m=1}^{k_i} \delta_{ijm} Z_{im}^{(t)} \right) \tag{2.10}$$

2.2 Latent Community Structure for λ

Direct estimation on λ may work well on smaller, fully ranked data sets, but a few problems arise otherwise. First, scaling to larger networks would be an issue as the number of parameters n^2 increases exponentially. This leads to computational inefficiency and potentially overfitting the model. Second, if we set $a = 1$ and $b = 0$ in (2.8) so that we are effectively finding the MLE, direct estimation of λ_{ij} is only possible if i has ranked j. Hence, we are interested in a lower dimensional representation of λ that accounts for these issues, namely a latent community representation:

$$\lambda_{ij} = \sum_{c=1}^{p} w_{ic} w_{jc} \tag{2.11}$$

where $c = 1, ..., p$ are the communities in the network and w_{ic} is the level of affiliation between individual i and community c. Now we are only estimating

n x p parameters. This representation has an additional benefit of providing some sense of community structure. Substituting λ_{ij} as in (2.11) and following the same steps up to (2.6) gives the following complete likelihood:

$$L_c(w) = \prod_{i=1}^{n} \prod_{m=1}^{k_i} \sum_{c=1}^{p} w_{ic} w_{\rho_{im} c} \exp(-z_{im} \sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_{c=1}^{p} w_{ic} w_{jc}) \tag{2.12}$$

To perform inference on w, we first want to write (2.12) in a Γ form so the prior will be conjugate and produce a Γ posterior. The first summation term in (2.12) will pose a problem as there is no way to isolate w_{ic} for optimization. We remove it by introducing another latent variable:

$$V_{ij} \sim (\frac{w_{i1} w_{j1}}{\sum_c w_{ic} w_{jc}}, ..., \frac{w_{ip} w_{jp}}{\sum_c w_{ic} w_{jc}})$$

with:

$$\mathbb{P}(V_{ij} = c) = \frac{w_{ic} w_{jc}}{\sum_{c=1}^{p} w_{ic} w_{jc}}$$

This can be interpreted as the probability of individuals i and j belonging to community c relative to them belonging to any of the other communities. It can also be seen as their proportion of shared affiliation with community c. Taking the likelihood of V and directly multiplying into (2.12) gives the new complete likelihood:

$$\begin{aligned} L_c(w) = & \prod_{i=1}^{n} \prod_{j \in \rho_i} \prod_{c=1}^{p} \frac{(w_{ic} w_{jc})^{\mathbb{1}\{v_{ij}=c\}}}{\sum_{c=1}^{p} w_{ic} w_{jc}} \\ & \times \prod_{i=1}^{n} \prod_{m=1}^{k_i} \sum_{c=1}^{p} w_{ic} w_{\rho_{im} c} \exp(-z_{im} \sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_{c=1}^{p} w_{ic} w_{jc}) \end{aligned} \tag{2.13}$$

Now we can see that the product over $j \in \rho_i$ is the same as the product over $m = 1$ to k_i. The ρ_{im} individuals for $w_{\rho_{im} c}$ are simply a reordering of the j individuals in ρ_i for w_{jc}. Hence, these terms cancel out and we are left with:

$$L_c(w) = \prod_{i=1}^{n} \prod_{j \in \rho_i} \prod_{c=1}^{p} (w_{ic} w_{jc})^{\mathbb{1}\{v_{ij}=c\}} \times \prod_{i=1}^{n} \prod_{m=1}^{k_i} \exp(-z_{im} \sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_{c=1}^{p} w_{ic} w_{jc}) \tag{2.14}$$

It is not possible to perform inference on both w_{ic} and w_{jc} at the same time. We therefore need to optimize each of the w_{ic} by component, treating all other parameters not equal to w_{ic} as a constant. We can then cycle through the w_{ic}'s, updating each parameter one at a time. To do this, we first introduce a new variable α_{ic} in terms of the indicator function $\mathbb{1}\{v_{ij} = c\}$ so the complete likelihood is rewritten as:

$$L_c(w) = \prod_{i=1}^{n} \prod_{c=1}^{p} w_{ic}^{\alpha_{ic}} \prod_{i=1}^{n} \prod_{m=1}^{k_i} \exp(-z_{im} \sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_{c=1}^{p} w_{ic} w_{jc}) \tag{2.15}$$

where:

$$\alpha_{ic} = \left(\sum_{j \in \rho_i} \mathbb{1}\{v_{ij} = c\} + \sum_{j | i \in \rho_j} \mathbb{1}\{v_{ji} = c\} \right) \tag{2.16}$$

is the number of individuals j nominated by i that share the same community plus the number of times i nominated by j share the same community. Note that v_{ij} takes on an integer value from 1 to p. Moving the product over $m = 1, ..k_i$ to a summation in the exponent, and rearranging the summations gives:

$$L_c(w) = \prod_{i=1}^{n} \prod_{c=1}^{p} w_{ic}^{\alpha_{ic}} \prod_{i=1}^{n} \exp(-\sum_{c=1}^{p} w_{ic} \sum_{m=1}^{k_i} (\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} z_{im} w_{jc})) \tag{2.17}$$

We move the summation over $c = 1, ...p$ out of the exponent down to a product so the products of the two arguments are common. Taking the log gives the complete log-likelihood:

$$\ell_c(w) = \sum_{i=1}^{n} \sum_{c=1}^{p} [\alpha_{ic} \log w_{ic} - w_{ic}\beta_{ic}] \tag{2.18}$$

which is now in a Γ form where we set:

$$\beta_{ic} = \sum_{m=1}^{k_i} \Bigg(\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} z_{im} w_{jc} \Bigg) \tag{2.19}$$

The Q function to derive the EM algorithm is:

$$Q(w, w^*) = \mathbb{E}_{V,Z|\rho,w^*}[\ell_c(w)] + \log p(w)$$

where $p(w)$ is a Gamma prior on w_{ic}, similar to (2.7). Note that we can take the expectations with respect to Z and V separately as they are independently distributed. Expanding this gives:

$$\begin{aligned} Q(w, w^*) = &\sum_{i=1}^{n} \sum_{c=1}^{p} \left[\mathbb{E}_{V|\rho,w^*}[\alpha_{ic}] \log w_{ic} - w_{ic} \mathbb{E}_{Z|\rho,w^*}[\beta_{ic}] \right] \\ &+ \sum_{i=1}^{n} \sum_{c=1}^{p} [(a-1) \log w_{ic} - b w_{ic}] \end{aligned}$$

where:

$$\begin{aligned} \mathbb{E}_{V|\rho,w^*}[\alpha_{ic}] &= \left(\sum_{j \in \rho_i} \mathbb{P}(v^*_{ij} = c) + \sum_{j | i \in \rho_j} \mathbb{P}(v^*_{ji} = c) \right) \\ &= \left(\sum_{j \in \rho_i} \frac{w^*_{ic} w^*_{jc}}{\sum_c w^*_{ic} w^*_{jc}} + \sum_{j | i \in \rho_j} \frac{w^*_{ic} w^*_{jc}}{\sum_c w^*_{ic} w^*_{jc}} \right) \end{aligned}$$

$$\mathbb{E}_{Z|\rho,w^*}[\beta_{ic}] = \sum_{m=1}^{k_i} \left(\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \frac{w_{jc}}{\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_{c=1}^{p} w^*_{ic} w^*_{jc}} \right)$$

Using these expectations and taking the argmax with respect to w_{ic} gives the update step:

$$\begin{aligned} w_{ic}^{(t)} =& \left(a - 1 + \sum_{j \in \rho_i} \frac{w_{ic}^{(t-1)} w_{jc}^{(t-1)}}{\sum_c w_{ic}^{(t-1)} w_{jc}^{(t-1)}} + \sum_{j | i \in \rho_j} \frac{w_{ic}^{(t-1)} w_{jc}^{(t-1)}}{\sum_c w_{ic}^{(t-1)} w_{jc}^{(t-1)}} \right) \\ & \times \left[b + \sum_{m=1}^{k_i} \left(\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \frac{w_{jc}^{(t)}}{\sum_{\substack{j \neq i \\ j \notin \rho_{<im}}} \sum_c w_{ic}^{(t-1)} w_{jc}^{(t-1)}} \right. \right. \\ & \left. \left. + \sum_{j \neq i} \left(\frac{w_{jc}^{(t)} \delta_{jim}}{\sum_{\substack{r \neq j \\ r \notin \rho_{<jm}}} \sum_c w_{jc}^{(t-1)} w_{rc}^{(t-1)}} \right) \right) \right]^{-1} \end{aligned} \tag{2.20}$$

where:

$$\delta_{ijm} = \begin{cases} 1 \text{ if } i \notin \rho_{<jm} \\ 0, \text{ otherwise} \end{cases}$$

is the indicator that i is not ranked higher than m by individual j, similar to how it is defined in (2.8). Note that this notation is equivalent to dropping δ_{jim} and summing the last argument over $j \neq i$ and $j | i \notin \rho_{<jm}$. We choose to proceed with the notation in (2.20). This last argument in the denominator seems to appear suddenly, but can be explained intuitively. Suppose we are finding the argmax for w_{11} and that individual 1 is well known and ranked by many other individuals. Hence, the term w_{11} will appear in the numerator of the last argument a smaller number of times than other less popular individuals due to the δ_{jim} condition. This results in a relatively higher updated value of w_{11}, as expected. Also note that the term $w_{jc}^{(t)}$ is updating simultaneously with every newly updated w_{ic}. For instance, suppose for iteration t we first update w_{11} using all other w_{-ic} i.e. parameters not equal to w_{11} from the previous iteration $t-1$. When we move on to update w_{12}, we use the updated w_{11} from time t and all other w_{-ic} from $t-1$. We do this until we have cycled through all w_{ic}. Terms labeled with $t-1$ use *only* the previous iterations parameters for the entire update step. An MCMC algorithm follows from this as:

For $i = 1...n, m = 1...k_i$:

$$Z_{im}^{(t)}|\rho, w^{(t-1)} \sim \varepsilon\left(\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}}\sum_{c=1}^{p} w_{ic}^{(t-1)}w_{jc}^{(t-1)}\right) \tag{2.21}$$

For $i = 1...n, j \in \rho_i$:

$$V_{ij}^{(t)}|\rho, w^{(t-1)} \sim \left(\frac{w_{i1}^{(t-1)}w_{j1}^{(t-1)}}{\sum_c w_{ic}^{(t-1)}w_{jc}^{(t-1)}}, ..., \frac{w_{ip}^{(t-1)}w_{jp}^{(t-1)}}{\sum_c w_{ic}^{(t-1)}w_{jc}^{(t-1)}}\right) \tag{2.22}$$

For $i = 1...n, c = 1...p$:

$$w_{ic}^{(t)}|\rho, w_{-ic}^{(t)}, Z^{(t)}, V^{(t)} \sim \Gamma\Bigg(a + \sum_{j\in\rho_i}\mathbb{1}\{V_{ij}^{(t)} = c\} + \sum_{j|i\in\rho_j}\mathbb{1}\{V_{ji}^{(t)} = c\}, \\ b + \sum_{m=1}^{k_i}\Bigg(\sum_{\substack{j\neq i \\ j\notin\rho_{<im}}} Z_{im}^{(t)}w_{jc}^{(t)} + \sum_{j\neq i} Z_{jm}^{(t)}w_{jc}^{(t)}\delta_{jim}\Bigg)\Bigg) \tag{2.23}$$

2.3 Hyperparameters

We briefly discuss the hyperparameters a and b of the Γ prior. Aside from being a conjugate prior to the likelihood, Γ reflects the assumption that the affiliation levels are positive [8]. The choice of hyperparameters determines the shape of the prior, where a large a and small b will result in a high mean and high variance, and vise versa for large b and small a. If we set $a = 1$ and $b = 0$, we get the MLE. We do not have an exact interpretation of the hyperparameters themselves, but we could set them depending on how varied we expect the community affiliations to be. Ultimately, the affiliations can be interpreted relative to each other.

The parameter b, however, is simply a scaling parameter on w_{ic} and is likelihood invariant. Hence, there is not much use in setting a prior for it [3]. In this case, we could set b as some function of a and vary a to see how it affects the log-likelihood. Alternatively, we could estimate the

hyperparameters from the data, similarly to how Caron and Doucet [3] do by using a Metropolis-Hastings random walk in conjunction with the MCMC sampler to accept or reject proposals of a at every iteration t. For this analysis, however, we do not consider this estimation of the hyperparameters and instead fix them from the start.

Another hyperparameter to consider is the number of communities p. While similar methods could be developed to estimate p, we do not explore them and instead set p based on existing knowledge and expectations of the network.

3 Simulated Data

We first test the model on simulated data. The data is simulated by fixing a and b and sampling w_{ic} from the resulting prior. We then use w_{ic} to compute λ_{ij} from (2.11), which we use as the rate parameters for Y_{ij}. We sample from each of the Y_{ij} to get arrival times and order them to produce the ranked lists. We then apply the model to see if we can recover the parameters. The data is formatted so that each row i represents the preferences of individual i. The columns are the rank of the preference, and the entries are the index of other individuals j. Hence, a directed edge exists from i to j if j is in row i. More details and code for simulating data are given in the Appendix B. Unless otherwise stated, we use R for all the modeling and simulations.

3.1 Parameter Recovery

We consider a simplified case where $p = 1$, i.e., we do not consider there to be any distinct communities. Hence we are estimating w_i, which for now can be interpreted as a "popularity level" rather than community affiliation since more well known individuals are likely to be ranked more frequently. We could understand $\lambda = \sum_c w_i w_j$ to reflect two well known individuals likely being affiliated with each other. The EM update step simplifies to:

$$w_i^{(t)} = (a + n_i)\Bigg[b + \sum_{m=1}^{k_i}\Bigg(\frac{\sum_{\substack{j\neq i\\ j\notin\rho_{<im}}} w_j^{(t)}}{\sum_{\substack{j\neq i\\ j\notin\rho_{<im}}} w_i^{(t-1)} w_j^{(t-1)}} + \sum_{j\neq i}\left(\frac{w_j^{(t)}\delta_{jim}}{\sum_{\substack{r\neq j\\ r\notin\rho_{<jm}}} w_j^{(t-1)} w_r^{(t-1)}}\right)\Bigg)\Bigg]^{-1} \tag{3.1}$$

where n_i is the number of times i is nominated by any j. An MCMC sampler follows as before, though now it is only necessary to sample from Z and w, as $\mathbb{P}(V_{ij} = c)$ will always be 1.

For i = 1,...,n:

$$Z_{im}^{(t)}|\rho, w^{(t-1)} \sim \varepsilon\left(\sum_{m=1}^{k_i}\sum_{\substack{j\neq i\\ j\notin\rho_{<im}}} w_i^{(t-1)} w_j^{(t-1)}\right) \tag{3.2}$$

$$w_i^{(t)}|\rho, w_{-i}, Z^{(t)} \sim \Gamma\Bigg(a + 1 + n_i, b + \sum_{m=1}^{k_i}\Bigg(\sum_{\substack{j\neq i\\ j\notin\rho_{<im}}} Z_{im}^{(t)} w_j^{(t)} + \sum_{j\neq i} Z_{jm}^{(t)} w_j^{(t)}\delta_{jim}\Bigg)\Bigg) \tag{3.3}$$

We set $a = 2$ and $b = 3$ and simulate the top $k = 8$ preference for 50 individuals. We run the MCMC sampler in (3.2) and (3.3) for 20,000 iterations with 2,000 burn-in samples. Diagnostics in **Figure 1** show the lag-1 autocorrelation and traceplots of the recovered parameter with the highest mean. We see the autocorrelation tends to 0 and the traceplot shows no obvious signs of increasing or decreasing patterns, suggesting the Markov chain is mixing reasonably well, and to an extent, we are sampling independent samples from the posterior distribution. Autocorrelation and traceplots of the other parameters show similar patterns.

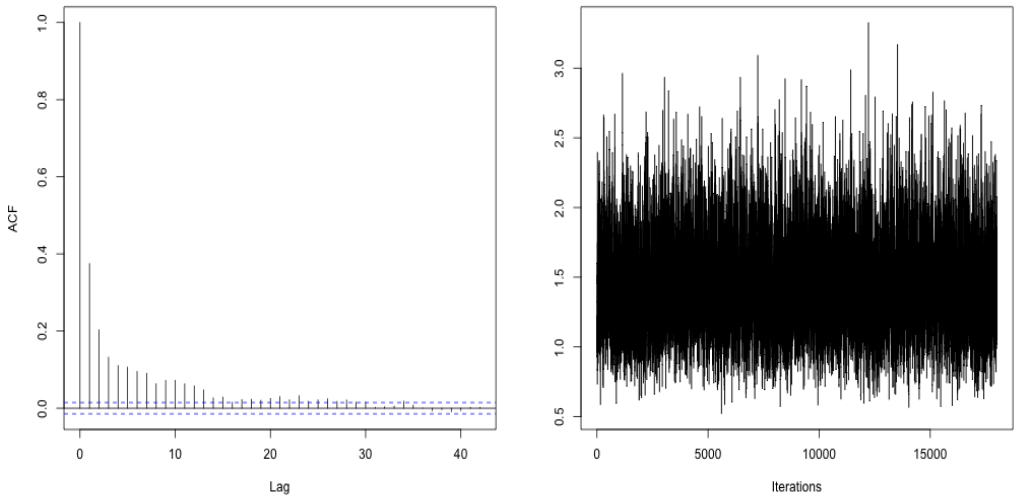


Figure 1: Lag-1 autocorrelation and traceplot of w_{42}, the recovered parameter with the highest mean. Note that the iterations are excluding burn-in.

In **Figure 2**, we look at the marginal posterior distributions of the highest and lowest simulated parameters to assess recovery over the range of w. We give the prior and posterior distributions of the recovered parameter, the point estimate of the recovered parameter, and the simulated parameter value. We see the sampler is recovering the parameters reasonably well, albeit slightly underestimated for the largest value and slightly overestimated for the smallest. Due to randomness when sampling from Y, individuals with the highest simulated parameters are more likely to be frequently ranked, though it could be the case they do not end up being ranked as well as we expect. Individuals with the lowest simulated parameter values are even more difficult to recover, as they may not end up being ranked at all. These issues can alter the parameter estimates, but they should still be within a reasonable range given the data, as we observe in this case. Simulating larger networks or more preferences could lead to better parameter recovery, though we do not explore this here.

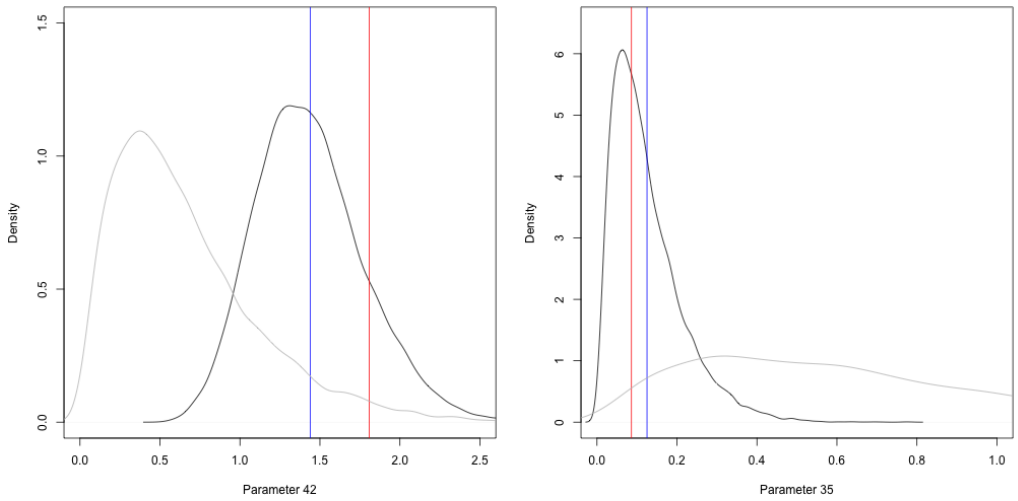


Figure 2: Marginal posterior distributions (black), prior distributions (grey), point estimates of recovered parameter (blue), and "true" parameter value (red) of w_{42} and w_{35}, the highest and lowest simulated parameters respectively.

3.2 Community Detection

We also create a small network with two clearly distinct communities and perform soft clustering to explore the community detection capabilities of the model. We do not use any sophisticated technique to form the network. We simply create two disconnected networks represented by two tables each with the same number of individuals and use random number generation to "insert" sensible (individuals cannot nominate themselves or the same person twice) directed edges within the networks. We then manually insert edges between some individuals across the two communities to create one connected network. We do this for 20 individuals, with individuals 1-10 in community 1 and the rest in community 2, and set $k = 4$ and $p = 2$. The network is visualized in **Figure 3**. We set $a = 2$ and $b = 3$ and initialize w with random draws from a Γ distribution. We run the EM algorithm in (2.20) with the same a and b until the parameters are not changing beyond a threshold of 0.0001. We assign an individual to a community depending on which of their w_{ic} parameters is greater, though consider an individual could be classified to either community if both of their parameters are similar, hence a soft classification. In this case, we do not pay much attention to the actual parameter value and are primarily interested in the classification. The

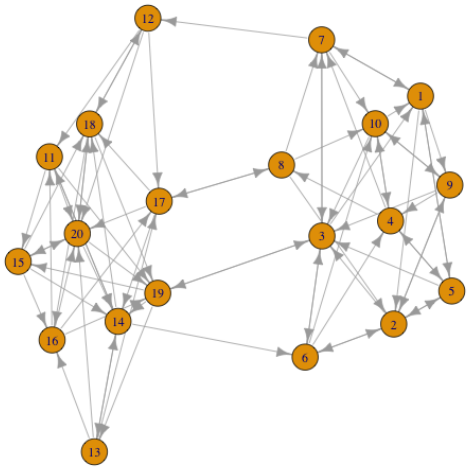


Figure 3: Simulated network with two distinct communities to test clustering capabilities.

EM algorithm returns the correct classification for all 20 individuals. We run it multiple times with more extreme initializations, such as a Γ distribution at $a = 1$ and $b = 100$ and vise versa and still obtain the same classifications for a fixed a and b of the EM algorithm.

One thing to note is the importance of the initial w. If we initialize w so all entries are the same value, the EM algorithm cannot distinguish communities and will reach a local maximum where each c for a given i is the same estimate. Furthermore, the algorithm does not consistently classify the first 10 individuals to community 1 and the latter 10 to community 2; sometimes it is reversed. Hence, we are not able to simply label the communities and have the algorithm return the clustering according to the labels. We are therefore inferring which community the algorithm is assigning an individual to based on the other individuals in that community and comparing the communities relative to each other. Varying the number of edges between communities or the number of distinct communities could provide insight into *how well* this functions as a classifier. But as we are primarily interested in simply whether or not the algorithm can classify, we do not assess this. Finally, to check that the EM algorithm is working correctly, **Figure 4** shows the log-likelihood is increasing at each iteration until it converges to a local maximum.

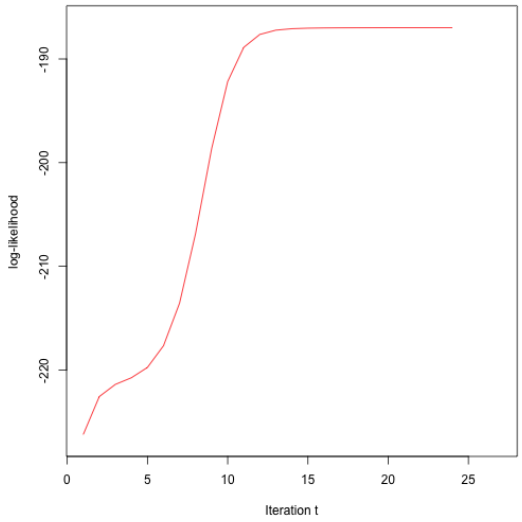


Figure 4: Log-likelihood at each iteration of the EM algorithm run on the network in **Figure 3**. The algorithm converges fairly quickly.

4 Application to Sampson's Monastery Data

We apply the model, specifically the EM algorithm, to Sampson's Monastery data with two goals: predicting or recommending further preferences for individuals who do not provide a complete list and classifying individuals into distinct communities. Note that we must take different approaches to clustering and prediction, where we use the entire network and all available preferences for clustering, but truncated preference lists for prediction. Hence, these tasks happen separately.

The data was collected by Samuel Sampson during his stay as an observer at a monastery to study the interactions and relationships between 18 incoming novice monks, where he asked each of the monks to rank three other monks in a variety of relations such as most and least liked, and positive and negative influence [14]. We know how to interpret each of these differently, but the model treats all relations as "top k" lists. We focus on the three most liked relation. Although it was recorded at 3 points in time, we use the latest. The network is given in **Figure 5**, with the original adjacency matrix in Appendix A.

Based on his analysis, Sampson originally classified the monks into 3 groups: Young Turks, Loyal Opposition, and Outcasts. Amid conflict, a

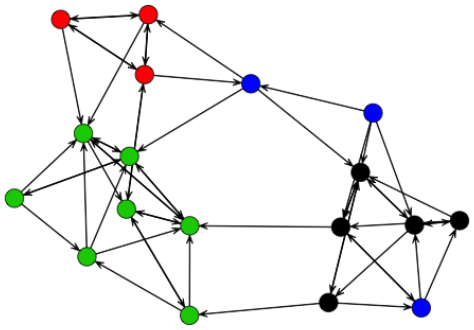


Figure 5: Network of Sampson's data where monks list their top 3 most liked connections, indicated by directed edges. Each vertex color represents a different group within the monastery: Young Turks (green), Loyal Opposition (black), Outcasts (red), and Waverers (blue). Note that the 3 Waverers were originally 2 Loyals and an Outcast.

fourth group, Waverers was formed [1]. Therefore we run the model for $p = 3$ and $p = 4$. We first convert the data from an adjacency matrix to the table format described in section 3. As mentioned in section 2, we do not do any hyperparameter estimation on the prior, and instead fix $a = 2$ and $b = 3$. This results in a Γ prior for the group affiliations with a mean of $2/3$ and variance of $2/9$. The rationale is that the network is small, and we assume the monks were in close proximity of each other given that they are in a monastery. So while we expect there to be clear communities according to Sampson's observations, we do not expect the community affiliations to vary greatly.

4.1 Predicting Further Preferences

All 18 monks listed their top 3 preferences, with 2 listing their top 4. We truncate this data by randomly choosing 6 monks and removing their third preference, or their fourth if they listed one. The other 13 monks have their full set of preferences. We run the EM algorithm on this truncated data,

Monk	Pred. Preference	Probability	True Preference
1	2	0.126	12
4	9	0.112	6
5	9	0.114	9
11	4	0.156	14
13	4	0.101	7
15	1	0.117	7

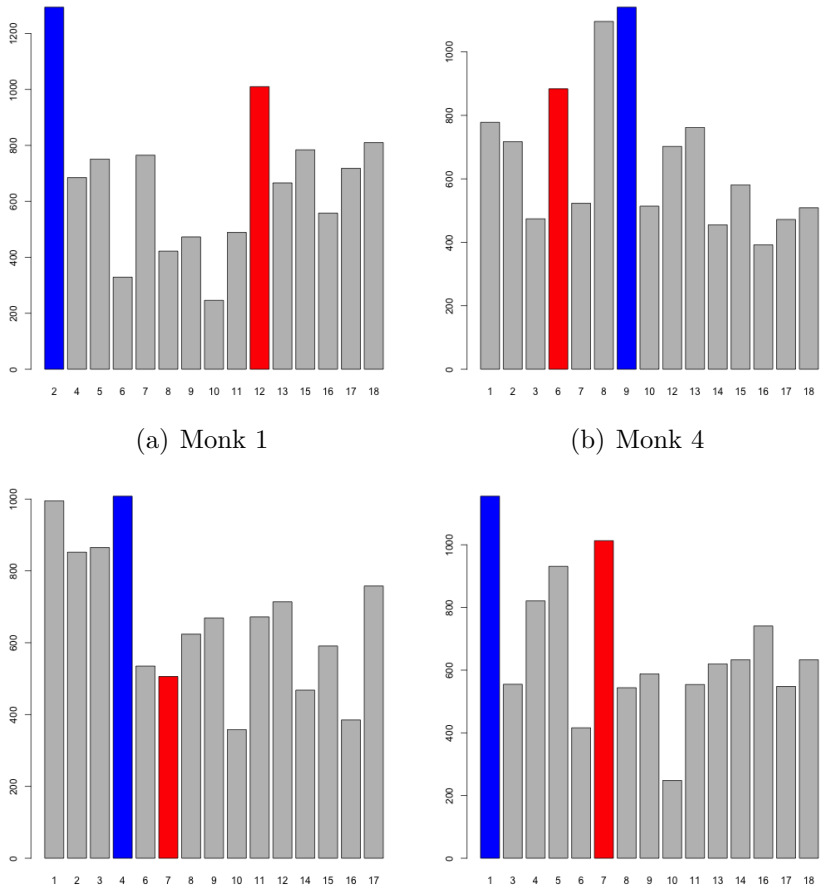
Table 1: Predicted vs true next preferences for the randomly chosen monks. The probabilities are calculated at $p = 4$. We see the model exactly predicts 1/6 preferences.

initializing w with values drawn from a uniform distribution between 0 and 1. To make predictions for the next preference m of monk i, we calculate the estimated affiliations $\hat{\lambda_{ij}}$ from (2.11) and choose the highest $\hat{\lambda_{ij}}$ where $j \notin \rho_i = (\rho_{i1}, .., \rho_{im-1})$. The probability of this preference follows as the highest $\hat{\lambda_{ij}}$ divided by the sum of the $\hat{\lambda_{ij}}$ that have not yet been ranked by i. We give the predicted next preference, the probability of that preference, and the true next preference in **Table 1**. The predictions are the same for both $p = 3$ and $p = 4$. This is not unexpected since all other parameters are the same, so the scale of the community affiliations will be the same, but we sum over an extra term for $p = 4$.

Choosing the highest $\hat{\lambda_{ij}}$ gives a point estimate for the predicted preference of monk i, but given the small size of the network and limited preferences, the point estimates can easily be influenced by popular monks, variability in the lists, and monks choosing others who are outside their group. Therefore, it is of more interest to see the distribution over potential preferences, given by:

$$\hat{\rho_{im}} \sim \left(\frac{\hat{\lambda_{ij}}}{\sum_{j \neq i} \hat{\lambda_{ij}}};\ j \neq \rho_{i1}, ..., \rho_{im-1} \right) \tag{4.1}$$

This suggests a few likely options rather than a single preference and provides a better assessment of the performance of the model than the point estimate. We can now think of this as recommending likely further connections, or predicting the top set of individuals each monk is likely to rank. From this point forward, we use "predicted" and "recommended" interchangeably to refer to likely preferences. For each monk, we construct a discrete distribution



(a) Monk 1

(b) Monk 4

(c) Monk 13

(d) Monk 15

Figure 6: Distribution of next preferences for monks 1, 4, 13 and 15 with their predicted (blue) and true (red) preferences indicated. The x-axis is the index of the possible monks to choose from and the y-axis is the frequency when sampled 10,000 times according to (4.1).

with probabilities according to (4.1) and draw 10,000 samples. To look at different groups, we show the histograms in **Figure 6** for monks 1, 4, 13, and 15. According to Sampson, they are Young Turks, Loyal Opposition, Outcast turned Waverer, and Young Turks, respectively. It is clear why it is not ideal to simply take the point estimate, as the true preference is typically among

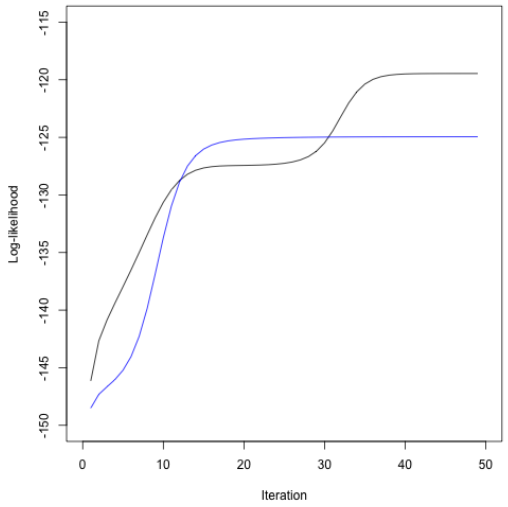


Figure 7: Log-likelihood of the model under $p = 3$ (black) and $p = 4$ (blue) over 50 iterations. The model with 4 groups converges quicker, but to a lower log-likelihood.

the top 2 or 3 potential preferences. The exception is monk 13, which can be explained by the fact that he chose an Outcast and Young Turk as his first two preferences, then a Loyal Opposition as his third. This inconsistency in groups is difficult for the model to capture. It does, however, choose a monk from the correct group. Monk 13 also illustrates how some predicted preferences can be swayed by popularity, especially if their given preferences do not indicate any clear affiliations; his top 2 predicted preferences are leaders of the Young Turks and Loyal Opposition [1].

4.2 Clustering Into Groups

For clustering, we initialize w with values from a standard uniform distribution and run the EM algorithm on the entire network at $p = 3$ and $p = 4$ up to a 0.0001 threshold. We plot the log-likelihoods in **Figure 7**. Even though the model under 3 groups produces a slightly higher log-likelihood, we choose to proceed with 4 groups to see if it can capture the Waverers. The parameter estimates and clusters are given in **Table 2**. We see the model is perfectly recovering the true clusters for the original 3 groups, with no monks being classified to the Waverers. We look at the parameter estimates for monks 8,

Monk	Waverers	Turks	Outcasts	Loyals	Cluster	True Group
1	0.309	0.6507	0.4916	0.1496	T	T
2	0.2902	1.213	0.3752	0.1524	T	T
3	0.1924	0.1283	0.8201	0.098	O	O
4	0.2413	0.1374	0.1641	1.1048	L	L
5	0.2825	0.1533	0.1963	1.2867	L	L
6	0.1968	0.1187	0.1392	0.6354	L	L
7	0.2689	1.0322	0.2085	0.1486	T	T
8	0.2172	0.1262	0.1465	0.6444	L	L/W
9	0.2692	0.203	0.173	0.8042	L	L
10	0.1501	0.0738	0.1065	0.2726	L	L/W
11	0.2375	0.1457	0.1548	0.5755	L	L
12	0.2969	1.1807	0.2235	0.1888	T	T
13	0.2712	0.1874	0.4122	0.2491	O	O/W
14	0.2827	0.4395	0.21	0.1682	T	T
15	0.2217	0.5996	0.1571	0.1221	T	T
16	0.1889	0.4802	0.1406	0.1097	T	T
17	0.1942	0.1563	0.6966	0.1148	O	O
18	0.225	0.1715	0.8679	0.1301	O	O

Table 2: Community affiliation parameter estimates and clusters given by the EM algorithm vs the true groups. Monks 8, 10, and 13 formed the Waverers group. The model correctly classifies each monk to their original group from the parameter estimates, Young Turks (T), Loyal Opposition (L), Outcasts (O), and Waverers (W).

10, and 13 who form the Waverers to see if their estimates suggest they could be classified to that group. For all three of them, their Waverers affiliation is the second highest, though this is the case for a majority of the monks.

One interesting thing to note is the clear separation between the Young Turks and Loyal Opposition, where those with their highest affiliation to the Young Turks often have their lowest with the Loyal Opposition and vise versa. During Sampson's stay, there was conflict over ideas and practices between the two groups [14]. The polarization in the estimates suggests the model is reflecting that dynamic.

5 Discussion

The results suggest that the model is experimentally capable in both predicting a set of preferences and identifying underlying community structure. There are, however, some limitations to consider for further applications, especially when applying the model to larger, more mixed networks.

For hyperparameters, we make the simple assumption of low variability in the community affiliations to fix a and b and know ahead of time what to set for p. As mentioned in section 2.3, these can all be estimated from the data using an MCMC sampler that updates the hyperparameters at each iteration according to a criteria. Though computationally more expensive, this would be useful when applying the model to social networks where proximity between individuals and community structure are unknown.

Though we address the goals of prediction and clustering separately, they are clearly connected in that the community structure affects the predicted preferences. For instance, monk 4 and 15 are Loyal Opposition and Young Turk respectively. In both cases, their top 3 predicted preferences are in the same group as them. In that same regard, the model has some difficulty capturing the cases when an individual is not strongly affiliated with a single community and has connections across communities. Furthermore, their predicted preferences are highly influenced by well-known individuals, as illustrated with monk 13 in section 4.1. The model is inclined to predict these well-known individuals simply on account of them having on average higher community affiliations from being nominated more often. As a result, two lesser-known individuals may be affiliated with each other, but the model will likely not predict the connection.

This issue reflects one property of the Plackett-Luce model in which the probability of an individual j with a low affiliation to i has too low a probability of being ranked by i [7]. Perhaps the most intuitive way to address this would be by restricting w such that $\sum_{c=1}^{p} w_{ic} = 1$, i.e., the community affiliations of each individual must add up to 1. However, this would cause information identifying well-known individuals to be lost, which, despite biasing some predictions, is still insightful to the network structure.

6 Conclusion

The Plackett-Luce model, a multiple comparison extension of the Bradley-Terry model, has commonly been used to analyze ranking data. We applied this model to a social network constructed from a FRN survey scheme, where individuals in the network are asked to list their top preferences among the other individuals. We introduced latent variables and derived EM and MCMC Gibbs samplers for parameter estimation using techniques similar to Caron and Doucet [3]. Under fixed hyperparameters, we applied the EM algorithm to a real network and showed it is capable of both accurately recommending further connections and capturing underlying network structure via clustering.

Acknowledgments

Thank you to Professor Francois Caron for his advice and supervision throughout this dissertation.

References

[1] S. Boorman, R. Breiger, and H. White. Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, 81(4):730–780, 1976.

[2] R. Bradley and M. Terry. Rank analysis of incomplete block designs. I. The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[3] F. Caron and A. Doucet. Efficient bayesian inference for generalized bradley-terry models. *Journal of the Computational and Graphical Statistics*, 21:174–196, 2012.

[4] A.P Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[5] P. Diaconis. Group representations in probability and statistics. In *IMS Lecture Notes*. Hayward, CA: Institute of Mathematical Statistics, 1988.

[6] B. Fosdick, P. Hoff, K. Stovel, and A. Volfovsky. Likelihoods for fixed rank nomination networks. *Network Science*, 1:253–277, 2013.

[7] I. Gromley and T. Murphy. Exploring voting blocs with the irish electorate: A mixture modeling approach. *Journal of the American Statistical Association*, 103: 1014–1027, 2008.

[8] J. Guiver and E. Snelson. Bayesian inference for plackett-luce ranking models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 377–384. New York, NY, 2009.

[9] K.M. Harris and J.R. Udry. *The National Longitudinal Study of Adolescent to Adult Health (Add Health)*. Chapel Hill, NC: Carolina Population Center, University of North Carolina-Chapel Hill; Ann Arbor, MI: Inter-university Consortium for Political and Social Research, 1994–2008.

[10] D. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32:384–406, 2004.

[11] J. Liu. *Monte Carlo Methods for Scientific Computing*. New York: Springer-Verlag, 2001.

[12] R. Luce. *Individual Choice Behavior: A Theoretical Analysis*. New York: Wiley, 1959.

[13] R. Plackett. The analysis of permutations. *Applied Statistics*, 24:193–202, 1975.

[14] S.F. Sampson. *A Noviate in a Period of Change. An Experimental and Case Study of Social Relationships*. PhD thesis, Cornell Univeristy, 1968.
[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]