An Extended Mixture of Experts Model for Rank Data: Insights from Irish Elections

Ben Heskin

Supervised by Dr. James Ng

A Final Year Project submitted in partial fulfilment of the requirements for the
degree of
Bachelor of Arts in Mathematics and Economics

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics
Trinity College Dublin
Ireland
April 2025

Declaration

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have completed the Online Tutorial in avoiding plagiarism 'Ready, Steady, Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Student Number: 21364899
Signed: Ben Heskin
Date: July 1, 2025

Abstract

Ranked Data arises when judges are asked to order a finite set of items, and it is present in various fields, such as health economics, information retrieval, and election studies. Often, judges only provide partial rankings of the finite set of items, be it by survey design or indifference to certain items after a particular preference level. It is often also true that judges can be subdivided into different clusters based on the both their ranking and social covariates via Mixture Models for ranked data. This project proposes a unified approach at the intersection of both of these realities, namely a Bayesian Mixture of Experts Model for Partially Ranked Data. The proposed model shows encouraging initial applications to election data from the 1997 Irish Presidential Election.

Contents

1 Introduction

Ranked data arises whenever judges are asked to give an ordering (which henceforth will be assumed an ascending ordering) of a finite set of items by some sort of preference metric. In many real world settings, judges only provide partial rankings. Moreover, subgroups of judges frequently exist, with their preferences shaped not only by latent taste clusters but also by observable covariates such as age, gender, or socioeconomic status.

The most common parametric approach in the literature to model ranked data is the Plackett-Luce model. The framework has various extensions, the Benter ranked data model that assumes that judges provide rankings at lesser certainty at each preference level, and the Plackett-Luce model for partially ranked data, which takes into account partial rankings. Yet, the Plackett-Luce models and its predecessors assume a homogeneous population of judges. Mixture modelling approaches have been introduced to attempt to explicitly account for latent clusters in ranked data. Two models in particular are of note - the model of Mollica and Tardella who account for partially ranked data [1] in their mixture model, and the Mixture of Experts model of Gormley and Murphy that lets the component mixing weights vary with covariates[2]. That said, no model exists within the literature that explicitly models partially ranked data while also considering covariates. As a result, the aims of this thesis are twofold.

- Firstly, the models in the literature will be discussed and replicated. Alongside this, proofs of key formulae and statements that were either terse in nature or not present in the literature are elicited in this work.

- Secondly, the pitfalls and oversights on the models will be considered and an attempt to rectify them will be made in a hybrid model formulated by the author of this project.

The report proceeds as follows:

- **Sections 2 & 3:** Give a background on the development of ranked data models, and defines the models used in this report.

- **Section 4:** Defines mixture models and the mixture of experts model.

- **Section 5:** Lays out common inference methods used in statistical analysis

- **Section 6:** Gives an overview of inference tricks used in this project.

- **Section 7:** Desribes the Mollica and Tardella mixture model for partially ranked data

- **Section 8:** Describes the Gormley and Murphy model Mixture of Experts model for ranked data.

- **Section 9:** Introduces the hybrid Mixture of Experts model for partially ranked data first proposed in this report.

- **Section 10:** Presents the results of a case study on opinion poll data.

- **Section 11:** Gives conclusions and suggests areas of further study.

2 Ranked Data

Ranked Data has uses in countless fields where it is required to rank a set of items, conforming to some criterion. Some examples of such fields include health economics [3], market research [4], and betting [5]. When discussing 'Ranked Data', it is important to formally define what ranked data explicitly is and the assumptions that will be made throughout this report.

Definition 2.1 (Ranked Data). Let $\{i_1, i_2, \ldots, i_J\}$ be a finite set of J items. A judge i submits an ordering, denoted by R_i, of some or all of these items. Such an ordering constitutes *ranked data*.

A particularly familiar use of ranked data in an Irish context is in elections. In every election, voters in Ireland are provided with a ballot with various people on it, and are asked to rank them in descending order of preference i.e. they place a number 1 in the box beside the candidate they prefer the most, a 2 in the box beside the candidate they prefer second, and so on. The voter continues this process until there are either no candidates left to give a preference to - providing a full ranking - or they are indifferent to who is elected out of the candidates - providing a partial ranking.

Figure 2.1: *A sample ballot from the Meath West constituency in the 2024 Irish General Election. This voter provides a full ranking of the available items. They prefer the Fianna Fáil candidate the most, the Labour Party candidate second, and so on.*

It should be emphasised that an ordering may not include every item in the choice set. This could occur either because the judge is genuinely indifferent among items beyond a certain rank, or because

the survey or polling design explicitly requests only the top choices of the respondent. For example, a market survey might ask respondents to name their three favourite restaurants in their city, while an election exit poll may collect only voters' top four candidate preferences. In the first instance, providing a complete and accurate ranking of all restaurants in a city would be impractical for respondents. In the second, capturing all preferences from voters could impose substantial administrative burdens on polling companies.

The statistical modelling of ranked data has been studied formally in the literature for nearly a century [6]. Initially, modelling assumed that there is some underlying modal ordering of items that follows some objective scale. When a set of rankings are elicited from a group of judges, it is presumed that the overall preferences of the group converges to this true modal ordering. Such an interpretation was established by Thurstone in 1927. A Thurstonian model assumes that when a person compares two items, say i and j, they are actually comparing latent utilities μ_i and μ_j, but with random error. The model posits that latent utilities for each item that are normally distributed, and that the probability that item i is preferred to item j is:

$$\mathbb{P}(i \succ j) = \Phi\left(\frac{\mu_i - \mu_j}{\sqrt{2}\sigma}\right)$$

where Φ is the the normal cumulative distribution function [7].

Rather than assuming each judge makes their ranking using a population wide heuristic that leads to a common overall preference and trying to model the processes that a judge undergoes to produce these rankings, later models take the orderings and attempt to find a description of the of the population based on the sample.

Later developments model the ranking process as an iterative choice system, where the judge makes $J(J-1)/2$ pairwise comparisons of the J available items. In his 1950 paper, Smith posits that the judge, item by item, reveals which item they prefer out of that item and all of the other items. Smith's early model laid the groundwork by focusing on the interpretation of a complete ranking as a collection of pairwise comparisons. The model indicates that the probability of a ranking R_i being declared by judge i is the product of the probabilities of each item being preferred over the item that follows it in the ranking, i.e.:

$$\mathbb{P}(R_i) = C \times \prod_{\substack{(a,b) \\ R_i(a) < R_i(b)}} p_{ab}$$

where p_{ab} is the probability that item a is preferred to item b, and C is a normalising constant that is chosen to ensure that the probabilities sum to 1 [8].

Building on this framework, Bradley and Terry formalised a parametric model that assigns each item a parameter, where θ_j represents the 'strength' of item b [9]. Hence, the probability that item a is preferred to item b can be represented as:

$$\mathbb{P}(a > b) = \frac{\theta_i}{\theta_i + \theta_j}$$

Recall that p_{ab} in the Smith model represents the probability that item a is ranked over item b. The Bradley-Terry model introduces a structured form that can be used to make inference and estimations for the strength of each item, by substituting in their above form for $\mathbb{P}(a > b)$ into the equation seen in Smith's work.

The Bradley-Terry model can be extended further. Their framework relies on paired comparisons, which can be inefficient to elicit over all J items in the choice set, especially if J is particularly large. Hence, procedures that treat the strength parameters as the probability of item j being ranked first over all of the remaining $J - 1$ items, can be more efficient. One such extension is the Plackett-Luce model for ranked data, which is discussed in the next section.

3

3 Parametric Models for Ranked Data

3.1 Plackett-Luce Model for Rank Data

In elections that use a ranked-choice-voting system, it is unreasonable to assume that all candidates or parties will receive the same support from voters. Each candidate typically receives varying levels of preference from the electorate. Many jurisdictions deploy a Ranked Choice Voting structure for their elections, allowing voters to give a ranking to some or all of the available candidates. Consequently, a ranked-data model must be utilised in the parametric study of such elections to accurately capture the support for the available candidates. In this section, the Plackett-Luce Model for Rank Data will be introduced, alongside an insight into its development and the limitations of the model's real world applications. Furthermore, the Benter Model for Rank Data is presented. An extension of the Plackett-Luce framework, the model's practical advantages over its predecessor are highlighted.

3.1.1 Development of the Model

Introduced in 1959, *Luce's Choice Axioms* provide the foundation for models explaining Ranked data. Luce axiomatically inferred that when an individual selects a particular item from a set, the probability of choosing the item is proportional to its perceived utility, relative to the sum of the utilities of all available options. He also established that these probabilities adjust proportionally when items are removed from the choice set.

Formally, the Luce Choice Axioms state, for $R \subset S \subset T$:

$$\mathbb{P}_T(R) = \mathbb{P}_S(R) \cdot \mathbb{P}_T(S) \tag{3.1}$$

$$\mathbb{P}_T(S) = \mathbb{P}_{T-\{x\}}(S - \{x\}) \tag{3.2}$$

$\forall\, x \in T$.

Here $\mathbb{P}_T(R)$ represents the probability of selecting item(s) R from set T. Although the first axiom is intuitive, that the total probability of choosing item(s) R from the superset T is a nested procedure across two subsets of T, it has an important extension. If $R, S \subset T$ and their intersection is empty, then $\mathbb{P}_T(R \cup S) = \mathbb{P}_T(R) + \mathbb{P}_T(S)$. Upon a repeated practice of this result, it can deduced that:

$$\mathbb{P}_T(S) = \sum_{x \in S} \mathbb{P}_T(x)$$

Substituting into (3.1):

$$\mathbb{P}_T(R) = \mathbb{P}_S(R) \cdot \sum_{x \in S} \mathbb{P}_T(x)$$

Which re-arranges to:

$$\mathbb{P}_S(R) = \frac{\mathbb{P}_T(R)}{\sum\limits_{x \in S} \mathbb{P}_T(x)} \tag{3.3}$$

Given that T is the superset of R and S, we interpret $\mathbb{P}_S(R)$ as the probability of picking subset R within subset S, and $\mathbb{P}_T(R)$ as the "worth" or utility of item(s) R within the superset T.

In his 1975 paper, Plackett uses the Luce probability ratio (3.3) to determine the probability of a certain group of horses finishing in the top three positions out of a wider pack of horses in a race [10]. Through sequentially withdrawing the already placed horses and re-normalising the remaining items, Plackett defines a general formula for the probability of permutation ijk for the top three places in a race:

$$p_{ijk} = p(i > j > k) = p_i \times \frac{p_j}{(1 - p_i)} \times \frac{p_k}{(1 - p_i - p_j)}$$

3.1.2 Formal Definition of the Plackett-Luce Model for Rank Data

Combining the work of Luce and Plackett, the Plackett-Luce distribution was developed. The concepts of the Plackett-Luce distribution have many applications and are often used to model rank data and the support for each candidate in elections. Analogous to the work of Plackett, probabilities are obtained through a string of sequential comparisons, where at each stage a single item is identified as the most preferred among those remaining and is then removed from further evaluation.

For a set of L items $\{\nu_1, \nu_2, \ldots, \nu_J\}$, the probability of ranking the j'th item first, under a Plackett-Luce model is:

$$\mathbb{P}_{PL}(\nu_j | \underline{\lambda}) = \frac{\lambda_j}{\sum_{\nu \in L} \lambda_i}$$

where λ_i is the worth or weight of item i. The probability of ranking the k'th item second given the j'th item was ranked first is:

$$\mathbb{P}_{PL}(\nu_k | \nu_j, \underline{\lambda}) = \frac{\lambda_k}{\sum_{i \in L \setminus \{\nu_j\}} \lambda_i}$$

i.e. the weight of the candidate already chosen is removed and the probabilities are re-normalised. The probability of a full ranking $\nu_1 > \nu_2 > \cdots > \nu_J$ is:

$$\mathbb{P}(\nu_1 > \nu_2 > \cdots > \nu_J | \underline{\lambda}) = \prod_{j=1}^{J} \frac{\lambda_{\nu_j}}{\sum_{\nu \in \{\nu_j, \nu_{j+1}, \ldots, \nu_J\}} \lambda_\nu}$$

In the context of elections, the weights λ_i are referred to as the **support parameters**, denoted by p_i, where $0 \le p_i \le 1$ and $\sum_{i \in L} p_i = 1$. These parameters describe the relative levels of support each candidate enjoys among the electorate.

The Plackett-Luce model also has the flexibility to define the probabilities of partial rankings. In scenarios where a judge has only revealed their top n_i preferences:

$$R_i = (r_{(i,1)}, r_{(i,2)}, \ldots, r_{(i,n_i)}) \implies \nu_1 > \nu_2 > \ldots > \nu_{n_i} > \text{(remaining unpreferenced items)}$$

is given by:

$$\mathbb{P}(R_i | \underline{\lambda}) = \prod_{l=1}^{n_i} \frac{\lambda_{r(i,l)}}{\sum_{j=1}^{J} \lambda_{r(i,j)} - \sum_{q=1}^{l-1} \lambda_{r(i,q)}},$$

where λ_{ν_k} is the support (or worth) parameter for item ν_k, and $r(i,k)$ represents the item that judge i ranked in the k'th position. As with the full ranking case, each step corresponds to selecting the next-most-preferred item among those still remaining, and then removing that item from subsequent stages, with the sequence ending not when there are no preferences left to give, but rather once the voter / consumer has no preference for any of the remaining items. The Plackett-Luce model has a wide range of applications, in fields such as politics, agriculture, consumer studies, and sports [11].

While effective and widely used, the Plackett-Luce model has limitations in Ranked Choice Voting applications. It assumes that all preferences are given with the same degree of certainty or care. For instance, in an election with 10 candidates, a voter may confidently rank their top choices but feel less strongly about their lower preferences, such as which candidates receive their 6th or 7th preference, and in what order. Hence, an extended model should be considered to better capture variations in the ballots present in elections that uses ranked-choice-voting.

3.2 Benter Model for Ranked Data

The Benter Model for Ranked Data addresses the limitation introduced in the last section. Benter's 1994 work was inspired by his efforts to arbitrage discrepancies between the probabilities implied by public betting odds and those predicted by his own Multinomial Logit Model in the context of horse race betting. His model systematically corrects for bias and refines the predictions by sequentially estimating multi-stage rankings, and accounts for the decreasing certainty associated with predicting lower finishing positions.

The concepts introduced by Benter can directly be applied to ranked-choice-voting models. In their 2008 work, Gormley and Murphy adapt a traditional Plackett-Luce Model for Rank Data to incorporate the uncertainty with which rankings at lower preference levels are given [2].

Definition 3.1. The probability of ballot $\underline{x}_i$, with the n_i rankings $x_1 > x_2 > \cdots > x_{n_i}$ from the list of J candidates $\{\nu_1, \nu_2, \ldots, \nu_J\}$ is derived to be:

$$\mathbb{P}(\underline{x}_i|\theta) = \mathbb{P}(\nu_1 > \nu_2 > \cdots > \nu_n | \, \underline{p}, \underline{\alpha})$$

$$= \frac{p_1^{\alpha_1}}{\sum_J p_i^{\alpha_j}} \cdot \frac{p_2^{\alpha_2}}{\sum_{J\setminus\{\nu_1\}} p_i^{\alpha_j}} \cdot \, \cdots \, \cdot \frac{p_{n_i}^{\alpha_{n_i}}}{\sum_{J\setminus\{\nu_1,\nu_2,\ldots,\nu_{n-1}\}} p_i^{\alpha}}$$

$$= \prod_{j=1}^{n_i} \frac{p_j^{\alpha_j}}{\sum_{i\in\nu_j,\nu_{j+1},\ldots,\nu_J}^{J} p_i^{\alpha_j}}$$

where p_i is the **support parameter** of candidate i.

The Benter Model for Rank Data introduces a **dampening parameter** $\underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. It reflects the certainty with which the electorate cast their $1'$st, $2'$nd, $\ldots n'$th preferences. Similar to the Plackett-Luce Model for Rank Data, the value of each of the dampening parameter α_j lies within the interval $[0,1]$. Additionally, the constraints $\alpha_1 = 1$ and $\alpha_n = 0$ are imposed to avoid over-parametrization.

4 Mixture Models

Data is frequently generated by heterogeneous populations where distinct subgroups follow different underlying distributions. In such cases, a single global model may fail to capture the complexity of the observed data. Mixture models address this challenge by representing the overall density as a weighted combination of component densities, each corresponding to an unobserved subgroup.

A further advancement in this framework is the *Mixture of Experts* (MoE) model. Unlike conventional mixture models, MoE models allow the mixing weights to depend on covariates. That is, the probability that an observation belongs to a particular component (or expert) is modelled as a function of observed characteristics of the input data. This conditional framework provides a more meaningful interpretation by explicitly linking covariates to subgroup membership.

4.1 Mixture Models

Data can often exhibit many underlying distributions, rather than a main overarching one. In such a case, the data follows a *multimodal distribution* which is characterised as a distribution with more than one mode or local 'peak'. When combined, these distributions form a *mixture model*, with each component distribution contributing to the overall combined density.
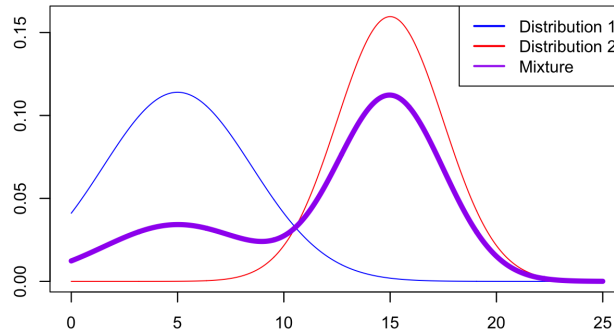
Figure 4.1: *Mixture of Two Normal Distributions*

Formally, the density of the observations $\underline{y}$, assuming that there are K underlying distributions, is:

$$\mathbb{P}(y_i) = \sum_{k=1}^{K} \pi_k \mathbb{P}(y_i|\theta_k)$$

where:

- π_k denotes the contribution (i.e. weight) of distribution k to the overall density,

- $\mathbb{P}(y_i|\theta_k)$ is the individual distribution of the k'th component.

Importantly, π_k is derived retrospectively from the posterior assignment of each output to the k'th cluster, meaning the weights are independent of the covariates or characteristics of the observed data. If each component represents a cluster of the observed data, there is no structured or model-based avenue to infer the factors influencing the inclusion of an individual observation in a specific cluster using a Mixture Model.

7

4.2 Mixture of Experts Model

The Mixture of Experts model, first introduced in 1991, extends the general Mixture Model framework. Initially developed as a method in computer science, its primary objective was to complete tasks more efficiently, typically handled by a single neural network with numerous parameters. This is achieved by dividing the process into smaller tasks, each assigned to a specific expert network trained and specialized in certain areas. The soft division of tasks is made by a **gating network,** with its weights determined by the covariates of the input data. This allows the model to dynamically determine the contribution of each expert based on the input.

The Mixture of Experts framework has various Computer Science applications, but one in particular stands out in 2025. Consider a Large Language Model (LLM) that a user inputs a text-based question to, and the model returns an appropriate answer. Instead of having one large neural network trained on each possible topic, a Mixture of Experts LLM gates the user's question efficiently to a subset of Expert Networks that are specifically trained (i.e. 'experts') in certain topics. For example, if the user asks a Statistics based question, the Gating Network would be expected to assign most of the tasks necessary to respond to an expert that has been trained on Statistics data, and maybe some of the tasks would be assigned to the Mathematics expert. Very few of the tasks, if any, would be assigned to the History expert. The model can be described as a conditional mixture model, as the

$$\mathbb{P}(\underline{x}_i, z_{ik} | \theta_k, \underline{\omega}_i) = \sum_{k=1}^{3} \pi_{ik}(\underline{\omega}_i) \mathbb{P}(x_i | \theta_k)$$

History Expert

Mathematics Expert

Statistics Expert

Gating Network

Stats Question

Demographics, constraints, bias vector...
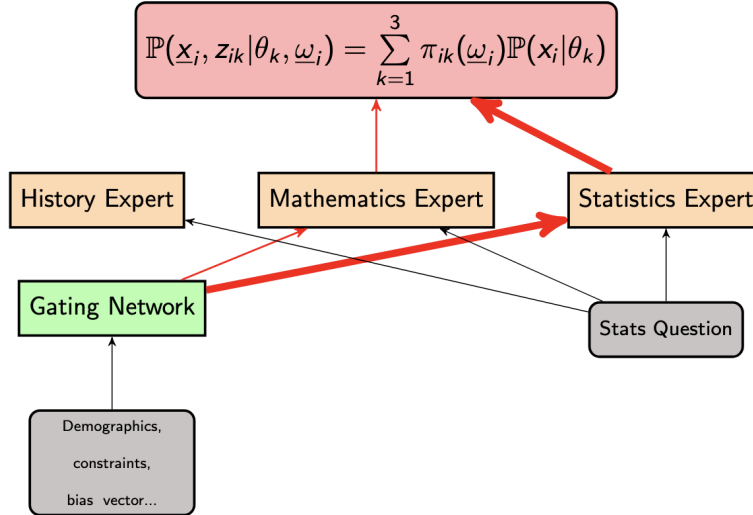
Figure 4.2: *A LLM example where the model is asked a Statistics question. The tasks are assigned efficiently amongst the 3 experts*

value of the weights (which are called gating networks) are now conditional on the observed data and any of its covariates [12]. The Mixture of Experts architecture has become increasingly popular in the Large Language Models field, due to its perception as an effective method for scaling model ability with minimal computational overhead. For example, DeepSeek's May 2024 'V2' LLM is a Mixture of Experts with 64 experts and 1.89 Billion parameters. That said, only 8 experts are ever activated in their model, using a framework called Sparse MoE, one of the latest developments in the field [13].

Although Mixture of Experts models were originally an innovation in the Computer Science field, a statistical interpretation can also be applied to the architecture. For M observations, where the

observed data $\boldsymbol{y}$ also has covariates or characteristics $\boldsymbol{x}$:

$$\mathbb{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \mathbb{P}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$$
$$= \sum_{i=1}^{M} \sum_{k=1}^{K} \mathbb{P}(z_{ik} = k|x_i) P(y_i|\theta_k)$$
$$= \sum_{i=1}^{M} \sum_{k=1}^{K} \pi_{ik} \mathbb{P}(y_i|\theta_k)$$

where:

- $\pi_{ik} = \pi_k(z_{ik} = k|\underline{x}_i)$ is the gating network i.e. the probability that observation i is a member of the k'th expert network, is given its covariates $\underline{x}_i$

- $\mathbb{P}(y_i|\theta_k)$ is the density of the k'th expert network, conditional on the parameters θ_k of that particular expert.

The gating network is derived using generalized linear models, typically multinomial linear regression, enabling a statistically structured incorporation of the covariates into the task division process. The statistical interpretation of the Mixture of Experts model is particularly useful in soft cluster-
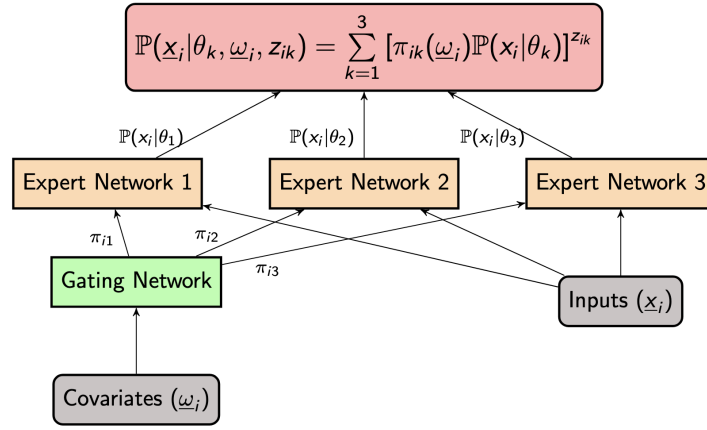
Figure 4.3: *An example of a Mixture of Experts configuration with 3 experts*

ing applications. By defining each expert as a data cluster, the model enables each observation to belong to more than one cluster while simultaneously incorporating covariates into the process. This provides a model-based framework for clustering, offering insight into why observations are assigned to specific clusters, in contrast to algorithmic methods that typically operate solely on the outcome variables and posterior observations.

From a statistical perspective, a Mixture of Experts model can have various functional forms. There are two sets of observed variables - the outcome variables $\boldsymbol{y}$ and their associated covariates $\boldsymbol{x}$. These observations can be modelled to explain the parameters in various ways:

a) **Mixture Model:**

The distribution of the output variables $\boldsymbol{y}$ is solely dependent on the latent mixture membership variables $\boldsymbol{z}$, and the parameters $\boldsymbol{\theta}$ attributed to each of these mixtures. They are independent of the covariates $\boldsymbol{x}$.

9

b) **Expert Network Mixture of Experts Model:**

The distribution of the output variables y is dependent on the latent expert network membership variables z, and the parameters θ attributed to each of these expert network. Now, the parameters θ are derived using the covariates x of each output variable, with the mixing weights π reaming independent of the covariates.

c) **Gating Network Mixture of Experts Model:**

The distribution of the output variables y is again dependent on the latent expert network membership variables z, and the parameters θ attributed to each of these expert networks. In this model, the parameters θ are derived independently of the covariates, and here the mixing weights π, defined by the gating network, are derived as a function of the covariates x.

d) **Full Mixture of Experts Model:**

In a Full Mixture of Experts model, both the expert and gating networks are influenced by the covariates x. This leads to both the expert network and gating network parameters, θ and π respectively, being a function of the covariates.
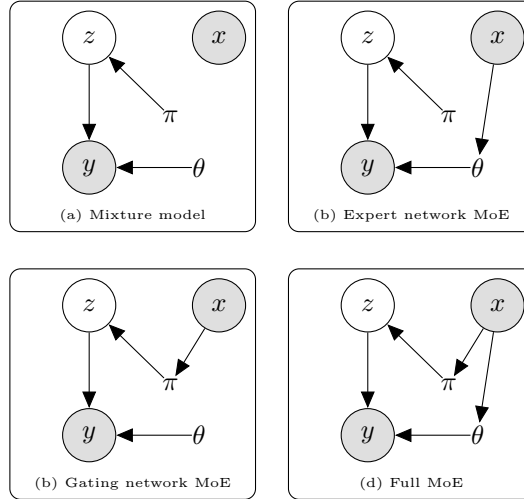
Figure 4.4: *Diagrams illustrating the four special cases of a Mixture of Experts model.*

The expert densities $\mathbb{P}(y_i \mid \theta_k)$ can represent underlying parametric distribution that best explain the unobserved heterogeneity in the data. An application that is the focus of this project is the use of Mixture of Experts models where the expert density is from the family of Plackett-Luce distributions. In such models, there are two sets of observed data - full or partial rankings R that judges reveal, alongside the social covariates x describing each judge - i.e. their age, gender, and socioeconomic status.

In such settings, there can typically be heterogeneity when modelling solely the ranked data. However, when analysing real-world data, additional heterogeneity often arises from observable attributes such as age, gender, or socioeconomic status. A gating network mixture of experts model addresses this challenge by linking these covariates directly to the probability of a judge being associated with a particular expert, without unnecessarily increasing the model's complexity. A study of the suitability of all four models represented in Figure 4.4 using data from the 1997 Irish Presidential Election empirically finds that using the Gating Network Mixture of Experts Model is best for the ranked and covariate data from an opinion poll taken during the campaign of that election [14].

5 Inference Methods

5.1 Maximum Likelihood Estimation

In statistical models, it is important to find the value of the underlying parameters ($\boldsymbol{\theta}$) that best explain the observed data. In other words, estimates of θ should be extracted that maximise the chosen *likelihood function* of the model over the parameter space [15].

Definition 5.1. For a continuous density function $f_Y(y; \theta)$ with unobserved parameters $\boldsymbol{\theta}$, for a random sample of n random variables, the *likelihood function* is written as:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f_Y(y_i; \theta)$$

To find the best fitting set of parameters, $\hat{\boldsymbol{\theta}}_{MLE}$ should be found such that $L(\boldsymbol{\theta}_{MLE}) \geq L(\boldsymbol{\theta})$ for all values of $\boldsymbol{\theta}$.

Definition 5.2. For the likelihood function of the random variable Y, the *Maximum Likelihood Estimation (MLE)* of the parameters is defined as:

$$\hat{\boldsymbol{\theta}}_{MLE} = \underset{\theta}{\operatorname{argmax}}\, L(\boldsymbol{\theta}; Y = y)$$

To be a valid *Maximum Likelihood Estimator,* the estimation $\hat{\boldsymbol{\theta}}_{MLE}$ must agree with the following properties:

1. **Consistency:** The estimator is consistent if, as the sample size n increases, $\hat{\boldsymbol{\theta}}_{MLE}$ converges in probability to the true parameter value i.e. :

$$\lim_{n \to \infty} \mathbb{P}(|\hat{\boldsymbol{\theta}}_{MLE} - \boldsymbol{\theta}| > \epsilon) = 0$$

   for every $\epsilon > 0$.

2. **Asymptotic Efficiency:** $\hat{\boldsymbol{\theta}}_{MLE}$ must be asymptotically efficient, meaning that in large samples of consistent estimators, $\hat{\boldsymbol{\theta}}_{MLE}$ achieves the minimum variance among all of these estimators.

3. **Asymptotic Normality:** After an appropriate rescaling, and as the sample size increases, the distribution of $\hat{\boldsymbol{\theta}}_{MLE}$ approaches the normal density.

4. **Parametrisation Invariance:** For $\hat{\boldsymbol{\theta}}_{MLE}$ and a bijective function g, the MLE of $g(\hat{\boldsymbol{\theta}}_{MLE})$ is $\widehat{g(\boldsymbol{\theta})}_{MLE}$.

[16]

5.2 Maximum a Posteriori Estimation

In Maximum Likelihood estimation, there is no consideration of any prior assumptions there may be for the parameters $\boldsymbol{\theta}$ of the model. Hence, an estimation procedure that includes information on the prior distribution of the parameters can be used. One such procedure is *Maximum a posteriori estimation,* extends MLE by combining the likelihood with a prior distribution $\mathbb{P}(\boldsymbol{\theta})$ that encapsulates any pre-existing beliefs about the parameter.

Definition 5.3. For the likelihood function of the random variable Y, the *Maximum a Posteriori Estimation (MAP)* of the parameters is defined as:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \underset{\theta}{\operatorname{argmax}} \left\{ L(X \mid \boldsymbol{\theta}) \mathbb{P}(\boldsymbol{\theta}) \right\}$$

MAP estimation regularises the likelihood of the observed data, leading to more robust estimates, particularly in environments with small sample sizes. Moreover, it facilitates Bayesian inference, since Bayes' theorem tells us that the posterior probability of the parameters $\boldsymbol{\theta}$ is proportional to the likelihood function of the data multiplied by the prior distribution of the parameters.

5.3 EM

The Expectation-Maximisation (EM) algorithm is an iterative procedure used to find local maximum likelihood estimates in models with latent variables (i.e., unobserved data). By alternately estimating the missing data (E-step) and optimising the model parameters (M-step), the EM algorithm guarantees a monotonic increase in the observed log-likelihood at each iteration.

5.3.1 Overview of the EM Algorithm

Suppose the observed data is represented by $\boldsymbol{X}$, and the latent variables are $\boldsymbol{Z}$. The EM algorithm follows two main stages:

0. *Initialisation Step (Optional):*

   The EM model only provides local Maximum Likelihood Estimates for the parameters. Sometimes, it is appropriate to use a simpler model or another reasonable method to initialise the parameter estimates.

1. *E-Step (Expectation Step):*

   In this step, compute the expectation of the complete-data log-likelihood with respect to the conditional distribution of the latent variables given the observed data and the current parameter estimates $\boldsymbol{\theta}^{(h)}$ from iteration h. This expected log-likelihood is defined as the Q-function:

   $$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(h)}) = \mathbb{E}_{\mathbf{Z}\mid\mathbf{X},\boldsymbol{\theta}^{(h)}}\big[\log \mathbb{P}(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\big].$$

   Here, the latent variables $\mathbf{Z}$ are "filled in" using the current estimate $\boldsymbol{\theta}^{(h)}$, effectively 'completing' the data.

2. *M-Step (Maximisation Step):*

   In the M-step, update the parameter estimates by maximizing the Q-function:

   $$\boldsymbol{\theta}^{(h+1)} = \arg\max_{\theta} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(h)}).$$

   This maximization guarantees that the observed data log-likelihood does not decrease, thereby ensuring that

   $$\mathcal{L}(\boldsymbol{\theta}^{(h+1)}) \geq \mathcal{L}(\boldsymbol{\theta}^{(h)}),$$

   where $\mathcal{L}(\boldsymbol{\theta})$ is the log-likelihood of the observed data.

The process is then repeated, with the newest estimates for the parameters calculated in the maximisation step of the previous iteration being used in the initialisation stage of the next iteration. The algorithm continues until it is deemed to be converged.

5.3.2 Convergence of the EM Algorithm

The Expectation-Maximisation algorithm only coverages linearly, meaning that in some cases results in slow convergence. Hence, simply noting that $|\mathcal{L}_{(h+1)} - \mathcal{L}_{(h)}|$ (the difference between the log-likelihoods of two consecutive iterations) is small can misleadingly lead to the conclusion that the algorithm has converged.

A more sensitive convergence diagnostic uses Aitken's acceleration criterion.It estimates the limiting log-likelihood:

$$\mathcal{L}_h^{\infty} = \mathcal{L}_{h-1} + \frac{1}{1 - c_h}(\mathcal{L}_h - \mathcal{L}_{h-1})$$

for $c_h \in (0, 1)$, where $c_h = \frac{\mathcal{L}_{h+1} - \mathcal{L}_h}{\mathcal{L}_h - \mathcal{L}_{h-1}}$.

Convergence is then declared when

$$|\mathcal{L}_\infty^{(h+1)} - \mathcal{L}_\infty^{(h)}|$$

falls below a predetermined threshold.

Sometimes, the $Q-$function is difficult to differentiate - especially when many parameters are being estimated simultaneously. Hence, methods often include an additional conditional step and a step that maximises a suitable surrogate function of the $Q-$function. Such concepts will be introduced and explained in the following sections.

5.4 MM Algorithms

5.4.1 Introduction

First formalised in 2000, the MM algorithm serves the purpose of replacing a hard to approximate function with a suitable smooth approximation surrogate function [17]. The method is considered a generalisation of the Expectation-Maximisation algorithm, freeing the parameter estimation process of the conditional expectation step [18].

The name *MM algorithms* stands for two processes: majorization-minimization and minorization-maximisation. In the former procedure, a function that majorizes the target function is derived. Formally, for the concave objective function $f(\boldsymbol{\theta})$, $g(\boldsymbol{\theta}|\boldsymbol{\theta}_m)$ minorizes f at the iteration θ_m if:

$$g(\boldsymbol{\theta}|\boldsymbol{\theta}_m) \geq f(\boldsymbol{\theta})$$

$$\& \ g(\boldsymbol{\theta}|\boldsymbol{\theta}_m) = f(\boldsymbol{\theta}_m)$$

For all θ_m. Similarly, $g(\boldsymbol{\theta}|\boldsymbol{\theta}_m)$ is a majorizing function of $f(\boldsymbol{\theta})$ if, for all θ_m:

$$g(\boldsymbol{\theta}|\boldsymbol{\theta}_m) \leq f(\boldsymbol{\theta})$$

$$\& \ g(\boldsymbol{\theta}|\boldsymbol{\theta}_m) = f(\boldsymbol{\theta}_m)$$
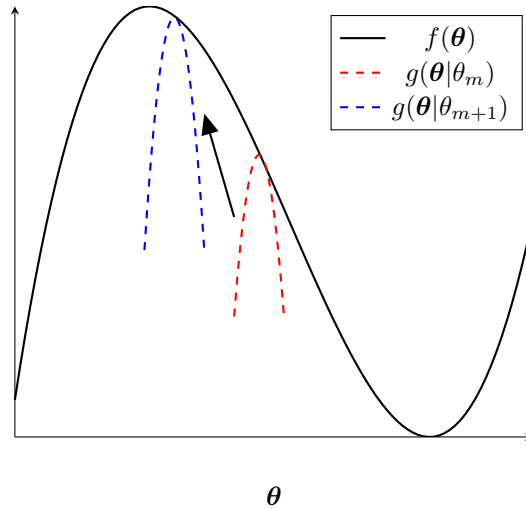
Minorization-Maximisation Illustration

5.4.2 Convergence

For a maximisation problem, the Minorization-Maximisation algorithm guarantees that at each iteration a monotonically increasing estimate for the parameters are produced [19]. At stage θ^m of the process, a surrogate function:

$$g(\boldsymbol{\theta}^m \mid \boldsymbol{\theta}^{(m-1)}) \leq f(\boldsymbol{\theta}^m)$$

is maximised to produce the new iterate at θ^{m+1} to produce $g(\boldsymbol{\theta}^{(m+1)} \mid \boldsymbol{\theta}^m)$, where:

$$g(\boldsymbol{\theta}^{(m+1)} \mid \boldsymbol{\theta}^m) \geq f(\boldsymbol{\theta}^m) \geq g(\boldsymbol{\theta}^m \mid \boldsymbol{\theta}^{(m-1)})$$

However, the surrogate g is constructed such that it is a lower bound on f, therefore:

$$f(\boldsymbol{\theta}^{m+1}) \geq g(\boldsymbol{\theta}^{(m+1)} \mid \boldsymbol{\theta}^m) \geq f(\boldsymbol{\theta}^m) \geq g(\boldsymbol{\theta}^m \mid \boldsymbol{\theta}^{(m-1)})$$
$$f(\boldsymbol{\theta}^{m+1}) \geq f(\boldsymbol{\theta}^m)$$

5.4.3 Construction of surrogate functions

Various methods are used in the construction of the surrogate functions. If the target is convex and differentiable, these properties can be exploited. Chiefly, the Mean Value Theorem states that for a differentiable function f there exists $c \in (x, y)$ such that:

$$f'(c) = \frac{f(y) - f(x)}{y - x}$$

Hence, if f is convex, that implies that f' is monotonically increasing i.e.:

$$f'(x) \leq f'(c) = \frac{f(y) - f(x)}{y - x}$$

which, after re-arranging is:

$$f(y) \geq f(x) + f'(x)(y - x)$$

The right hand side of this equation is therefore an appropriate minorizing function of f.

For a twice-differentiable convex function g, an adaptation of a Taylor expansion around y can be used to bound it from above with a quadratic approximation:

$$g(y) \leq g(x) + g'(x)^t(y - x) + \frac{1}{2}(y - x)^t \boldsymbol{B}(y - x)$$

defining a matrix $\boldsymbol{B}$ such that $\boldsymbol{H} < \boldsymbol{B}$, where $\boldsymbol{H}$ is the Hessian Matrix. This restriction ensures that the surrogate truly minorizes the target function as it guarantees that the term that introduces curvature in the surrogate approximation overestimates it [20]. Hence, a suitable majorizing function of g has been constructed.

6 Methods for Tractable Inference

Recall that for a Mixture of Experts model, the gating expert weights are often modelled using generalised linear models such as multinomial logistic regression functions:

$$\log\left(\frac{\pi_{ik}}{\pi_{i1}}\right) = \beta_{k0} + \beta_{k1}\omega_{i1} + \beta_{k2}\omega_{i2} + \cdots + \beta_{kL}\omega_{iL}$$

with π_{i1} being the baseline group of that model. After re-arranging, the probability π_{ik} of input i, with covariates $\underline{\omega}_i$, being a member of expert k is given by:

$$\pi_{ik} = \frac{e^{(\underline{\beta}_k^T \underline{\omega}_i)}}{\sum_{k'=1}^{K} e^{(\underline{\beta}_k'^T \underline{\omega}_i)}}$$

This softmax-like form lacks a closed form solution in Expectation-Maximisation procedures as the M-step requires solving a set of non-linear equations without analytical solutions. Also, such issues can be further compounded by complete or quasi-complete separation - where the outcome variable nearly perfectly or perfectly divides a predictor or a combination of predictors, which can lead to non-finite estimates [21].

The likelihood function for the Plackett Luce distribution poses similar issues. The probability of a full ranking of J items is given as:

$$\mathbb{P}(i_1 > i_2 > \cdots > i_J \mid \underline{\lambda}) = \prod_{j=1}^{J} \frac{\lambda_{i_j}}{\sum_{i \in \{i_j, i_{j+1}, \dots, i_J\}} \lambda_i}$$

where λ_i is the worth of each item i. In an EM algorithm, the M-step proves difficult. In this model, each possible subset of items must be considered at each iteration, meaning as the number of items that are ranked increases, the number of possible subsets being used as the normalising denominator increases quickly. Such a combinatorial representation of the likelihood function means that a closed form for the M-step is not achievable, with parameter estimation historically being conducted using computational techniques, such as the slower and more inefficient Newton-Raphson Method [22][23].

This section will introduce methods within both frequentist and Bayesian frameworks that can be employed for parameter estimation.

6.1 MM Algorithm Estimation

The MM-algorithm constructs surrogate functions of a difficult to optimise likelihood function, allowing for parameter estimation that was beforehand impossible with analytical methods. In the Minorization- Maximisation optimisation, a surrogate function that minorizes the original likelihood function is defined and maximised. If designed appropriately, the maximum of the surrogate function will coincide with the maximum of the original likelihood function. This approach proves a useful tool for parameter estimation in frequentist models.

6.1.1 MM Algorithm for Logistic Regression

For a logistic regression model with l categories of independent variables, the log- likelihood function is:

$$q = \sum_{l=1}^{L} \left[\exp(\underline{\beta}^T \underline{\omega}_l) - \log\left(\sum_{l=1}^{L} \exp(\underline{\beta}^T \underline{\omega}_l)\right) \right]$$

Direct maximization can be difficult, hence a minorization-maximisation approach is employed. Specifically, a local quadratic surrogate function around the previous iterations estimate of the gating parameters $(\underline{\beta}^{(h)})$ is constructed:

$$q(\underline{\beta}^{(h)}) + q'(\underline{\beta}^{(h)})^T(\underline{\beta}^{(h+1)} - \underline{\beta}^{(h)}) + \frac{1}{2}(\underline{\beta}^{(h+1)} - \underline{\beta}^{(h)})^T \boldsymbol{B}(\underline{\beta}^{(h+1)} - \underline{\beta}^{(h)})$$

A suitable matrix $\boldsymbol{B}$ should be chosen, where $\boldsymbol{B}$ bounds the Hessian Matrix $\boldsymbol{H}(\beta^{(h)})$ from below. For a logistic regression model, the probability $\pi_l(\beta)$ that the dependent variable $\overline{Y}$ is equal to 1 can be represented as

$$\ell(\boldsymbol{\beta}) = \frac{\exp(\beta^T \omega_l)}{1 + \exp(\beta^T \omega_l)}$$

The second partial derivative, with respect to the gating network parameters is:

$$\nabla^2 \ell(\boldsymbol{\beta}) = -\sum_{l=1}^{L} \pi_l(\beta) \left[1 - \pi_l(\beta)\right] \omega_l \omega_l^T$$

$\pi_i(\beta) \left[1 - \pi_i(\beta)\right]$ is maximised when $\pi_i(\beta) = \frac{1}{4}$, hence the negative definite matrix $\boldsymbol{B} = -\frac{1}{4} \sum_{i=1}^{M} \underline{\omega}_i \underline{\omega}_i^T$ is defined to derive the minorizing surrogate function:

$$q(\underline{\beta}_k^{(h)}) + q'(\underline{\beta}_k^{(h)})^T (\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) + \frac{1}{2}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})^T \left(-\frac{1}{4} \sum_{i=1}^{M} \underline{\omega}_i \underline{\omega}_i^T (\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) \right)$$

around the point $\underline{\beta}_k^{(h)}$. Now, this surrogate function is maximised:

$$\frac{\partial}{\partial \beta} \left[q(\underline{\beta}_k^{(h)}) + q'(\underline{\beta}_k^{(h)})^T (\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) + \frac{1}{2}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})^T \left(\boldsymbol{B}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) \right) \right] = q'(\underline{\beta}_k^{(h)}) + \boldsymbol{B}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})$$

$$\implies$$

$$\underline{\beta}_k^{(h+1)} = \underline{\beta}_k^{(h)} - \boldsymbol{B}^{-1} q'(\underline{\beta}_k^{(h)})$$

is the formula that defines the value of the next iteration, in terms of the estimate for the gating network parameters derived at the previous step.

6.1.2 MM Algorithm for a Plackett-Luce Model

Under an Expectation Maximisation framework, the log likelihood function for a Plackett-Luce density, with N items, is:

$$\mathcal{L}(\underline{p}) = \sum_{t=1}^{N} \left[\log p_t - \log \left(\sum_{s=t}^{N} p_s \right) \right]$$

with p_t being the support parameter of item t.

Here, the term $-\log \sum_{s=t}^{N} p_s$ is problematic as it may not always be differentiable, so a minorizing function is constructed. Given the MM algorithm ensures monotonically increasing estimates for the parameters, an appropriate lower bound is constructed around the most recent estimate of $\underline{p}$, with each element defined as $\overline{p}_s$.

Recall that, for a convex function $f(y)$, it can be bound below by:

$$f(y) \geq f(x) + f'(x)(y - x)$$

Where x is an appropriately chosen value to be a lower bound. In the case of the MM algorithm, the estimation of the target parameter from the previous iteration is a logical choice. The function to be optimised in this case is $-\log(y)$, which is convex. Therefore:

$$-\log(y) \geq -\log(x) + [\log(x)]'(y - x)$$

$$\geq -\log(x) + \frac{1}{y}(y - x)$$

$$\geq -\log(x) - \frac{y}{x}$$

16

or:

$$-\log \sum_{s=t}^{N} p_s \geq \log \sum_{s=t}^{N} \overline{p}_s + 1 - \frac{\log \sum_{s=t}^{N} p_s}{\log \sum_{s=t}^{N} \overline{p}_s}$$

with $\overline{p}_s$ representing the estimate for the support parameter from the previous iteration. In this inequality, the right hand side is a minorizing surrogate function of $-\log \sum_{s=t}^{N} p_s$. It follows that:

$$q \geq \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \left(\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} + 1 - \frac{\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}}{\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$

$$\geq \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \sum_{t=1}^{n_i} \left(\frac{\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}}{\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$

A further surrogate function is constructed to deal with the term $-\sum_{s=t}^{N} p_s$. It follows that the final surrogate function can be constructed as:

$$Q \geq q = \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \left(\frac{\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} p_{kc(i,s)}}{\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$

up to a constant.

6.2 Data Augmentation Approaches

Although data augmentation can be used in many fields, such as deep learning [24], this section will evoke a statistical interpretation. Data augmentation is a general group of methods that are often utilised to allow for a more efficient maximum likelihood estimation or maximum a posteriori, or even finding one that was otherwise impossible to achieve using more traditional approaches. In a frequentist framework, the most fundamental augmentation approach is the EM algorithm, where the hard to approximate posterior likelihood function is augmented with unobserved latent variables to ensure convergence to the true value of the parameters being modelled. On the other hand, the MM algorithms described in the previous section are not data augmentation processes as there are no latent variables produced.

The main focus of the extensions introduced in this project will be on methods in Bayesian frameworks. Within the Bayesian setting, data augmentation is especially useful to allow for a closed form representation of the posterior to use in gibbs sampling. If the augmented likelihood, when combined with the prior assumptions is of the same probability distribution family as the posterior, then the prior is called a *conjugate prior* for the likelihood, and the prior and posterior are conjugate to each other.

Example 6.1. Consider a Binomial likelihood with a Beta prior. The posterior is proportional to:

$$\mathbb{P}(\theta | \underline{\mathbf{x}}_i) \propto \mathbb{P}(\underline{\mathbf{x}}_i | \theta) \mathbb{P}(\theta)$$
$$\propto \theta^x (1-\theta)^{n-x} \theta^{\alpha-1} (1-\theta)^{\beta-1}$$
$$\propto \theta^{x+\alpha-1} (1-\theta)^{n-x+\beta-1}$$

which is still a conditional Beta distribution,:

$$\theta | x \sim \text{Beta}(\alpha + x, \beta + n - x)$$

If a prior is conjugate to a newly-augmented likelihood, difficult integration or the use of any intensive numerical methods are avoided, leaving us with a closed form posterior distribution for the parameters to either maximised or sample from.

6.2.1 Data Augmentation for Logistic Regression

For Binary response models, the probit regression model is easier to deal with in a bayesian framework, in contrast to the logistic regression model and its intractable likelihood function. In 1993, Albert and Chib introduced a latent variable augmentation approach to model probit functions, where the binary outcomes are treated as as "thresholded realisations" of an underlying normally distributed latent variable [25]. In 2013, Polson, Scott, and Windle introduced a class of distributions called Pólya-Gamma distributions. In their work, for logistic regression coefficients β with the prior assumption that they are normally distributed, are conjugate to a likelihood function that is augmented by Pólya-Gamma random variables.

Definition 6.2. A random variable ω has a Pólya-Gamma distribution with parameters $b > 0$ and $c \in \mathbb{R}$ if:

$$\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - \frac{1}{2})^2 + (\frac{c}{2\pi})^2}$$

where:

- $g_k \sim G(b, 1)$ are gamma random variables.

- $\stackrel{D}{=}$ denotes equality in distribution [26].

Polson et al's paper also derives a closed form for the expected value of a Pólya-Gamma variable $\omega \sim PG(b, c)$, which is a convenient fact to utilise in processes such as the EM algorithm.

Lemma 6.3. *The expected value of a Pólya-Gamma random variable ω, with parameters $b > 0$ & $c \in \mathbb{R}$:*

$$\mathbb{E}(\boldsymbol{\omega}) = \frac{b}{2c}\tanh(c/2)$$

Proof. First, recall that a Pólya-Gamma distributed random variable for the parameters $b > 0$ & $c \in \mathbb{R}$ are equal in distribution to:

$$\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - \frac{1}{2})^2 + (\frac{c}{2\pi})^2}$$

Hence, the random variable $\omega \sim PG(b, 0)$ can be defined as:

$$\omega \stackrel{D}{=} \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - \frac{1}{2})^2}$$

where the g_k's are independent gamma random variables with $g_k \sim \Gamma(b, 1)$. It is known that the Laplace transformation of such a random variable g_k can be written as:

$$\mathbb{E}[e^{-s \cdot g_k}] = (1 + s)^{-b}$$

if g_k is scaled by a constant i.e. $(y_k = 1/2\pi^2(k - \frac{1}{2})^2$ in this case), then the laplace transform becomes:

$$\mathbb{E}\left[e^{-t y_k}\right] = \left(1 + \frac{t}{2\pi^2 \left(k - \frac{1}{2}\right)^2}\right)^{-b}. \tag{6.1}$$

Note that each of the g_k's are independent by definition, hence the Laplace transformation is an infinite product over k. Also note that $\cosh(z)$ can be represented as an infinite product[27]:

$$\cosh(z) = \prod_{k=1}^{\infty} \left(1 + \frac{4z^2}{(2k-1)^2 \pi^2} \right)$$

hence,

$$\mathcal{L}_{PG(b,0)}(t) = \cosh(\sqrt{t/2}) = \prod_{k=1}^{\infty} \left(1 + \frac{t}{2\pi^2 \left(k - \frac{1}{2}\right)^2} \right)^{-b} = \left(\frac{1}{\cosh\left(\sqrt{\frac{t}{2}}\right)} \right)^{b}$$

Now, the $PG(b,c)$ density is obtained by re-weighting the $PG(b,0)$ distribution is *exponentially tilted* by a factor of $-c^2/2$, to have a valid density:

$$\mathcal{L}_{PG(b,c)}(t) = \frac{\mathcal{L}_{PG(b,0)}(t + \frac{c^2}{2})}{\mathcal{L}_{PG(b,0)}(\frac{c^2}{2})}$$

$$= \frac{\cosh^b(\frac{c}{2})}{\cosh^b \left(\sqrt{\frac{c^2/2 + t}{2}} \right)}$$

To get the expected value of this Laplace transform, we take the negative derivative evaluated at $t = 0$,. Given we have constructed the transform with a $-t$ term in the exponent, we have a Moment Generating Function, whose first derivative gives us the expected value:

$$\mathbb{E}(\omega) = \frac{b}{2c} \tanh(c/2)$$

$\square$

To the reader, it may not be immediately obvious why a Pólya-Gamma augmentation helps in finding a conjugate form for a logistic regression likelihood function. One key theorem of the authors' work sets up such a solution.

Theorem 6.4. *Let* $\omega \sim PG(b,0)$, $b > 0$. *Then :*

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = 2^{-b} e^{\kappa \psi} \int_0^\infty e^{-\omega \psi^2/2} p(\omega) d\omega$$

holds $\forall\, a \in \mathbb{R}$, *where* $p(\omega)$ *denotes the density as* $p(\omega) = PG(\omega \mid b, c = 0)$, *and* $\kappa = a - \frac{b}{2}$.

Proof. Begin by examining the expression on the left-hand side of the theorem. Note that its denominator can be decomposed to a term that reflects the hyperbolic cosine ratio:

$$\frac{e^{a\psi}}{(1 + e^\psi)^b} = \frac{(e^\psi)^a}{(e^{\psi/2}(e^{\psi/2} + e^{-\psi/2}))^b}$$

$$= \frac{(e^\psi)^a}{2^b e^{b\psi/2} \cosh^b(\psi/2)}.$$

$$= \frac{e^{a\psi}}{e^{b\psi/2}}\, 2^{-b} \cosh^{-b}(\psi/2)$$

$$= 2^{-b} e^{(a - b/2)\psi} \cosh^{-b}(\psi/2)$$

19

by introducing the Laplace transform of described in Lemma 6.3:

$$= 2^{-b} e^{\kappa \psi} \int_0^\infty e^{-\omega \psi^2 / 2} p(\omega) d\omega$$

for $\kappa = a - b/2$, as required. $\qquad \square$

Recall that the logistic regression likelihood is derived using the log likelihood of success p:

$$\log \left(\frac{p}{1-p} \right) = \boldsymbol{\beta}^T \boldsymbol{x}$$

which, after re-arranging, implies that the likelihood function is:

$$\mathcal{L}(\boldsymbol{\beta}) \propto \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{x})^{\sum y_i}}{1 + \exp(\boldsymbol{\beta}^T \boldsymbol{x})^N}$$

with the contribution of indicator variable i, with y_i the observed binary outcome for the i'th observation being:

$$\mathcal{L}_i(\boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}^T x_i)^{y_i}}{1 + \exp(\boldsymbol{\beta}^T x_i)}$$

If ψ is replaced with $\boldsymbol{\beta}^T \boldsymbol{x}$ in the Theorem 6.4, i.e. a linear function of predictor variables and their coefficients, the identity shows that the logistic likelihood, which was originally non-conjugate, can be written as a conditionally Gaussian likelihood in $\boldsymbol{\beta}$ given a latent Pólya-Gamma random variable ω.

Theorem 6.5. *Under a Pólya-Gamma augmentation, the likelihood of a logistic regression function can be expressed in a conditionally Guassian form.*

Proof. Note that:

$$\mathcal{L}_i(\boldsymbol{\beta} \mid \omega_i) = \frac{\exp(\boldsymbol{\beta}^T \underline{x}_i)^{y_i}}{1 + \exp(\boldsymbol{\beta}^T \underline{x}_i)}$$

$$\propto \exp(\kappa_i \boldsymbol{\beta} \, \underline{x}_i^T) \int_0^\infty \exp\{-\omega_i (\boldsymbol{\beta} \, \underline{x}_i^T)^2\} p(\omega_i)$$

$$= -2 \exp\{\kappa_i \boldsymbol{\beta}^T \underline{x}_i\} \times \mathbb{E}_{p(\omega_i \mid b=1, c=0)} [\exp(-\omega_i (\boldsymbol{\beta}^T \underline{x}_i)^2 / 2)]$$

$$\overset{\blacklozenge}{\propto} \exp\{\kappa_i \boldsymbol{\beta}^T \underline{x}_i\} \exp\{-\omega_i (\boldsymbol{\beta}^T \underline{x}_i)^2 / 2\}$$

$$= \exp \left\{ -\frac{\omega_i}{2} \left((\boldsymbol{\beta} \underline{x}_i)^2 - \frac{2\kappa_i}{\omega_i} (\boldsymbol{\beta} \underline{x}_i) \right) \right\}$$

$$\propto \exp \left\{ -\frac{\omega_i}{2} \left(\boldsymbol{\beta} \underline{x}_i - \frac{\kappa_i}{\omega_i} \right)^2 \right\}$$

where $\kappa_i = y_i - 0.5$ is the exponential tilting parameter and $p(\omega_i) \sim PG(1, 0)$. The step $\blacklozenge$ holds because the expectation with respect to ω_i, under the Pólya-Gamma distribution, returns a constant. The final line is a result of completing the square in $\boldsymbol{\beta}$. Combining the contribution of every independent variable i:

$$\mathbb{P}(\boldsymbol{\beta} \mid \boldsymbol{\omega}, \boldsymbol{y}) \propto \mathbb{P}(\boldsymbol{\beta}) \cdot \prod_{i=1}^{N} \mathcal{L}_i(\boldsymbol{\beta} \mid \omega_i)$$

$$\propto \mathbb{P}(\boldsymbol{\beta}) \cdot \exp\left\{ -\sum_{i=1}^{N} \frac{\omega_i}{2} \left(\boldsymbol{\beta} \underline{x}_i^T - \frac{\kappa_i}{\omega_i} \right)^2 \right\}$$

$$\propto \mathbb{P}(\boldsymbol{\beta}) \cdot \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} \left(\boldsymbol{\beta} \underline{x}_i^T - \frac{\kappa_i}{\omega_i} \right) \omega_i \left(\boldsymbol{\beta} \underline{x}_i^T - \frac{\kappa_i}{\omega_i} \right) \right\}$$

Define $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_N)$ and $z_i = \frac{\kappa_i}{\omega_i}$, where $z = (z_1, \ldots, z_N)^\top$:

$$\propto p(\boldsymbol{\beta}) \exp\left\{ -\frac{1}{2}(z - X\boldsymbol{\beta})^T \Omega (z - X\boldsymbol{\beta}) \right\}$$

which for a flat prior, β is a multivariate normally distributed random variable with mean $(X^T \Omega X)^{-1} X^T \kappa$ and the covariance matrix $(X^T \Omega X)^{-1}$, as required.

$\square$

Theorem 6.6. *Consider a logistic regression model with linear predictor*

$$\psi_i = \boldsymbol{\beta} x_i^T,$$

By augmenting the model with Pólya-Gamma latent variables ω_t, the complete-data log-likelihood can be expressed as

$$\mathcal{L}_C(\beta) = \sum_{i=1}^{N} \left[\kappa_i \, \psi_t - \frac{1}{2} \, \omega_i \, \psi_i^2 \right].$$

where:

$$\kappa_i = y_i - \frac{1}{2}.$$

Proof. For each indicator variable i, the logistic likelihood is given by

$$L_i(\boldsymbol{\beta}) = \frac{\exp(\psi_i)^{y_i}}{(1 + \exp(\psi_i))},$$

with $\psi_i = \boldsymbol{\beta} \underline{x}_i^T$. Defining

$$\kappa_i = y_i - \frac{1}{2},$$

a key identity from Pólya-Gamma data augmentation shows that

$$\frac{\exp(\psi_i)^{y_i}}{(1 + \exp(\psi_i))} \propto \exp(\kappa_i \, \psi_i) \int_0^\infty \exp\left(-\frac{1}{2} \, \omega_i \psi_i^2 \right) p(\omega_i \mid 1, 0) \, d\omega,$$

where $p(\omega_i \mid 1, 0)$ denotes the density of a Pólya-Gamma random variable with parameters 1 and 0.

In the context of an EM algorithm, ω_i are the latent variables. Therefore, the complete-data likelihood for indicator i becomes

$$\mathcal{L}_C(\boldsymbol{\beta} \mid \omega_i) \propto \exp\left\{ \kappa_i \, \psi_i - \frac{1}{2} \, \omega_i \psi_i^2 \right\} p(\omega_i \mid 1, 0)$$

Since $p(\omega_i \mid 1, 0)$ does not depend on $\boldsymbol{\beta}$, taking the logarithm and ignoring any additive constants yields the complete-data log-likelihood (for each observation)

$$\log \mathcal{L}_C(\beta, \omega_t) = \kappa_t \psi_t - \frac{1}{2} \omega_t \psi_t^2$$

up to a constant. Summing over all observations $i = 1, \ldots, N$ gives

$$\mathcal{L}_C(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left[\kappa_i \psi_i - \frac{1}{2} \omega_t \psi_t^2 \right]$$

$\square$

In MAP estimation, the prior distribution is included to regularise the estimate. A sensible prior for the logistic regression parameters $\boldsymbol{\beta}$ is a normal distribution i.e. $\beta \sim N(b, B)$, as it is clearly a member of the same distribution family as the likelihood function. In this case,

$$p(\beta \mid \omega, y) \propto \exp\left[-\tfrac{1}{2}(\beta - b)^T B^{-1}(\beta - b)\right] \times \exp\left[-\tfrac{1}{2}(z - X\beta)^T \Omega (z - X\beta)\right]$$

meaning that the posterior is proportional to a multivariate normally distributed random variable with mean $(X^T \Omega X + B^{-1} \cdot b)^{-1} X^T \kappa$ and the covariance matrix $(X^T \Omega X + B^{-1})^{-1}$ in a MAP setting [28]. Now, a well-defined statistical framework has been defined, namely the Pólya-Gamma augmentation of logistic regression functions. This method can be used to make Bayesian inference over the heretofore intractable likelihood form of these functions.[1]

Although originally applied to binary logistic regression, Pólya-Gamma augmentation can be extended to multinomial logistic outcome scenarios. Consider a multinomial response model with K categories. A vector of regression parameters $\boldsymbol{\beta}_k$ corresponds to each response category, where the log odds of the response is modeled by the dot product of this vector and a vector of i independent predictor variables $\boldsymbol{x}_i$. If the likelihood of $\boldsymbol{\beta}_k$ is conditioned on the latest parameter estimates for the other response categories, $\boldsymbol{\beta}_{-k}$, one can adapt the binary augmentation strategy by introducing a latent Pólya-Gamma random variable for each observation–category pair. In particular, by augmenting the likelihood with these latent variables, the contribution from each observation can again be expressed in a form that is conditionally Gaussian with respect to $\boldsymbol{\beta}_k$. Consequently, the complete-data likelihood for category k becomes conditionally conjugate to the normal prior on $\boldsymbol{\beta}_k$, allowing for closed form Bayesian inference. A formal derivation will be made in subsequent sections.

---

[1] Blog posts by Gregory Gundersen & Louis Tiao were also helpful in the derivations of this subsection.

6.2.2   Data augmentation for the Plackett Luce Distribution

Consider three independent exponential random variables:

$$X \sim \mathcal{E}(p) \qquad\qquad Y \sim \mathcal{E}(q) \qquad\qquad Z \sim \mathcal{E}(s)$$

Now, we discuss a novel way to calculate the probability that $X < Y < Z$. If we augment our interpretation of this probability, this calculation can be reimagined as a series of inter-arrival time computations, given the inter-arrival times of Poisson processes are exponentially distributed. Hence, for X to be less than Y, then the parameter p (i.e. X's inter-arrival time) must have a lower value than the parameter q. A similar line of argument allows us to conclude that q must have a lower value than s. The probability of such an event is the combination of two events occurring:

$$\mathbb{P}(X < Y < Z) = \mathbb{P}(\{X < \min(Y, Z)\}) \cap \mathbb{P}(Y < Z)$$

However, Poisson processes have the independent increments condition, meaning whatever happens at the first arrival point has no affect on what happens at the second, and so on. Hence:

$$\mathbb{P}(X < Y < Z) = \mathbb{P}(\{X < \min(Y, Z)\}) \times \mathbb{P}(Y < Z)$$

$$= \frac{p}{p + q + s} \times \frac{q}{q + s}$$

[29].This example unveils the property known as an *exponential race*. The similarity between this
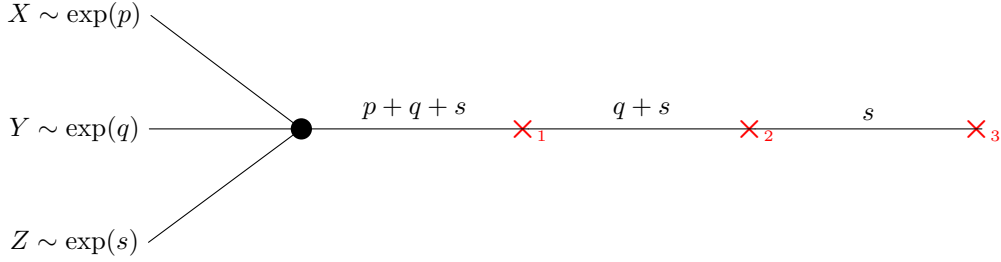

Figure 6.1: *A diagram visualising the event $X < Y < Z$. The calculation is reimagined as an exponential race, where the probability is thought of in terms of competing exponential random variables that remain at the i'th inter-arrival time $\times_i$.*

property and the Plackett-Luce distribution should not be lost on the reader. Indeed, if the exponential variables $\{X, Y, Z\}$ are replaced with Horse X, Horse Y, and Horse Z, who are associated with the exponential parameters or *winning probabilities* λ_X, λ_Y, and λ_Z respectively, then the probability of the permutation of a race result XYZ, i.e. the probability that X has the lowest arrival time, and so on, is:

$$\mathbb{P}(X \text{ first}, Y \text{ second}, Z \text{ third}) = \frac{\lambda_X}{\lambda_X + \lambda_Y + \lambda_Z} \times \frac{\lambda_Y}{\lambda_Y + \lambda_Z}$$

which is the exact result that Plackett derived for the same scenario. In a less statistical sense, the *exponential race* can be thought of a framework to get the probability of object / person / candidate x being the first to reach objective y. In the previous example, the probability that Horse X is the first to reach the finish line in a horse race is derived. Another use case could be calculating the probability of Candidate X being the first to reach the quota of votes necessary to win an election.

In 2010, Caron and Doucet efficiently introduce exponential data augmentation approaches for ranked data models [30]. Specifically for the Plackett-Luce model:

$$\mathbb{P}(\nu_1 > \nu_2 > \cdots > \nu_J \,|\, \underline{\lambda}) = \prod_{j=1}^{J} \frac{\lambda_{\nu_j}}{\sum_{\nu \in \{\nu_j, \nu_{j+1}, \ldots, \nu_J\}} \lambda_\nu}$$

the latent variables $\omega_{ij} \sim f_{\exp}(\sum_{k=j}^{n_i} \lambda_{r(i,j)})$, where $r(i,j)$ represents which item person i ranks in the j'th position, $f_{\exp}$ represents the density of the exponential distribution, and n_i is the number of items person i ranks. Caron and Doucet's approach is successful in attaining closed form updates for the Expectation-Maximisation algorithm, without using frequentist methods such as the MM algorithms, which were used in the last section. Their work was a fundamental step in the development of effective Bayesian ranked data models, as they were the first example of closed form updates for Gibbs-Sampling in the literature.

7 A Mixture Model for Partially Ranked Data

In 2017, Mollica and Tardella introduce a Bayesian Mixture Model for partially ranked data. When considering ranked data, it is not plausible that every person who is asked to provide a preference profile gives a full ranking of the M items available. This could be by design i.e. a marketing survey that asks respondents for their top three favourite brands, or an exit poll that only collects data for the top four preferences of voters in an election that uses ranked choice voting to mitigate administrative burden. Or indeed it may be a consequence of how the electoral system is set up. For example, in Irish Elections, there is no rule on how many preferences people give, often leading to situations where people give partial rankings as they truly are indifferent to who they support after revealing a certain amount of their rankings.

Even in scenarios where partial rankings are provided, there can still be inference to be made and heterogeneity present in the rank data. Mollica and Tardella's work is the first example in the literature where a finite mixture model is fitted to partially ranked data. Their model takes a Bayesian approach, which can be especially informative as the assumed priors help in stabilising the estimation, and a full picture is attained of the posterior density. The data augmentation approach used in their work efficiently handles the tricky normalisation structure of a partial ranking Plackett-Luce density, especially when various judges provide rankings of differing lengths [1].

7.1 Model Specification

In the model, it is assumed that the M rankings, $R_1, \ldots, R_i, \ldots R_M$, are conditionally sampled from a K mixture component mixture of Partial Plackett-Luce distributions:

$$R_1, \ldots, R_i, \ldots, R_M \mid \boldsymbol{p}, \boldsymbol{\pi} \sim \sum_{k=1}^{K} \left[\pi_{ik} \mathbb{P}_{PL}(R_i | \underline{p}_k) \right]^{z_{ik}}$$

where,

- R_i is the partial ranking provided by the ith judge.

- $\underline{p}_k = (p_{1k}, \ldots, p_{jk}, \ldots, p_{Nk})$ is the vector of support parameters for the N items in mixture k.

- $\underline{\pi}_k = (\pi_1, \ldots, \pi_k, \ldots, \pi_K)$ is the vector of the mixing weights for each of the K mixtures.

- z_{ik} is the latent variable such that:

$$z_{ik} = \begin{cases} 1 & \text{Judge } i \text{ is a full member of mixture } k, \\ 0 & \text{otherwise} \end{cases}$$

- $\mathbb{P}_{PL}(R_i \mid \underline{p}_k)$ is the density of a Partial Plackett-Luce ranking model:

$$\mathbb{P}(R_i | \underline{p}_k) = \prod_{t=1}^{n_i} \frac{p_t}{\sum_{j=1}^{N} p_{jk} - \sum_{q=1}^{t-1} p_q},$$

such that n_i the amount of preferences judge i reveals.

The normalisation term of the Partial Plackett-Luce density $(\sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q)$ is particularly difficult to maximise in EM algorithms, due to the coupling of the support parameters across all of the candidates and mixtures. In a Bayesian paradigm, can be applied to data augmentation can be applied to successfully yield a likelihood that is more tractable. In their model, Mollica and Tardella introduce an exponentially distributed latent variable, which considers a better representation for the normalisation term across the t preferences that voter i reveals:

$$y_{it} \sim \mathcal{E}(\lambda_{it} = \sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q)$$

with density:

$$f_{\exp}(y_{it}|\lambda_{it}) = \lambda_{it} \exp(-\lambda_{it} y_{it})$$

This form successfully partitions the support parameters $\boldsymbol{p}$ that are in the problematic normalisation term.

It should be noted that the probability mass function of the latent allocation variables $\boldsymbol{\pi}$ is multinomial in structure, namely:

$$\mathbb{P}_{\text{Multinomial}}(\mathbf{z} \mid \boldsymbol{\pi}) = \prod_{i=1}^{M} \prod_{k=1}^{K} \pi_k^{z_{ik}}.$$

For a fully Bayesian model, a prior distribution must be assigned to both the exponentially augmented Partial Plackett Luce and the multinomial likelihoods. The priors need to be conjugate to these likelihoods, and they should also be a plausible fit for the parameters of the model. We know that for a mixture model, the mixing weights must sum to 1, i.e. $\sum_{k=1}^{K} \pi_k = 1$. Hence, a suitable prior for $\boldsymbol{\pi}$, which is multinomial-distributed, is a Dirichlet distribution, with density:

$$\mathbb{P}_{\text{Dirichlet}}(\boldsymbol{\pi}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

$$= \frac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

$$\propto \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

as it not only assumes that the parameters sum to 1, it is also conjugate to the multinomial distribution:

$$\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{z}) \propto \mathbb{P}(\boldsymbol{z} \mid \boldsymbol{\pi})\mathbb{P}(\boldsymbol{\pi})$$

$$= \prod_{i=1}^{M} \prod_{k=1}^{K} \pi_k^{z_{ik}} \times \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

($b_k = \sum_{i=1}^{M} z_{ik}$ is introduced for notational convenience)

$$= \prod_{k=1}^{K} \pi_k^{b_k + \alpha_k - 1}$$

The last part of the derivation shows us that $\boldsymbol{\pi}$ is conditionally Dirichlet distributed:

$$\boldsymbol{\pi} \mid \boldsymbol{z} \sim \text{Dirichlet}(b_1 + \alpha_1, b_2 + \alpha_2, \ldots, b_K + \alpha_K)$$

Now, we need to define a suitable prior for the Plackett-Luce support parameters. Given that the likelihood has been augmented with exponentially distributed random variables, a natural choice is a Gamma prior:

$$p_{jk} \sim \text{Ga}(c_{jk}, d_k)$$

The Gamma distribution is defined only on the positive real numbers, which aligns with the requirement that support parameters must be positive. Additionally, the conjugacy between the Gamma prior and the exponential likelihood facilitates closed form updates during inference:

$$\mathbb{P}(\boldsymbol{p} \mid \boldsymbol{y}) = \prod_{t=1}^{n_i} \prod_{i=1}^{M} \prod_{k=1}^{K} \prod_{j=1}^{N} \left[L(p_{jk} \mid \boldsymbol{y}) \times \frac{1}{\Gamma(c)d^c} \cdot p_{jk}^{c-1} \exp(-dp_{jk}) \right]$$

The likelihood of p_{jk} is augmented by z_{ik} also, to give a latent variable for the EM algorithm used in the MAP estimation. Given $\lambda_{it} = \sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q$ the likelihood is given now by:

$$L(p_{jk} \mid \mathbf{y}) \propto p_{jk}^{\gamma_{jk}} \exp\{-p_{jk} \cdot S_{ij}\},$$

where,

- γ_{jk} is the effective count i.e. the number of times p_{jk} appears in the normalisation across the data, or more formally:

$$\gamma_{ik} = \sum_{i=1}^{M} z_{ik} u_{ij}$$

  with:

$$u_{ij} = \begin{cases} 1 & \text{if } j \in R_i \\ 0 & \text{otherwise} \end{cases}$$

- S_{ij} is the sum of the corresponding latent variables y associated with the support parameters:

$$S_{ij} = \sum_{t=1}^{n_i} \delta_{ijt} y_{it}$$

- δ_{itj} is defined to quantify if item j has not been chosen yet at preference level t, i.e. it is still available:

$$\delta_{itj} = \begin{cases} 1 & \text{if } j \notin \{r_{(i,1)}, \ldots, r_{(i,t-1)}\} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the completed likelihood function is multiplied by the prior to achieve a form that is conjugate to the posterior:

$$\mathbb{P}(\boldsymbol{p}|\boldsymbol{y}, \boldsymbol{z}) \propto p_{jk}^{c_{jk}+\gamma_{jk}-1} \exp\left\{-p_{jk}\left(d_k + S_{ij}\right)\right\}$$

forming a conditionally gamma form for the support parameters:

$$p_{jk} \mid \boldsymbol{y}, \boldsymbol{z} \sim \text{Ga}(c_{jk} + \gamma_{ik}, \ d_k + S_{ij})$$

7.2 Maximum a Posteriori Estimation

To learn the local posterior mode, Mollica and Tardella introduce an EM algorithm to optimise the posterior distribution. The EM algorithm relies on the Q function, which is iteratively maximised at the M-step with respect to the parameters being optimised. Here, we have two latent variables, y_{it} and z_{ik}. Their expected value is calculated at each iteration's E-step, based on the current estimate of the parameters, where:

$$\hat{y}_{it} = \mathbb{E}[y_{it}] = \frac{1}{\lambda_{it}} = \frac{1}{\sum_{j=1}^{N} \overline{p}_j - \sum_{q=1}^{t-1} \overline{p}_q}$$

$$\hat{z}_{ik} = \mathbb{E}(z_{ik}) = \frac{\overline{\pi}_k \mathbb{P}_{\mathrm{PPL}}(R_i \mid \overline{p}_k)}{\sum_{k'=1}^{K} \overline{\pi}_{k'} \mathbb{P}_{\mathrm{PPL}}(R_i \mid \overline{p}_{k'})}$$

with $\overline{p}$ and $\overline{\pi}$ representing the estimates for the parameters of the model at the last iteration of the EM algorithm. 'In this algorithm for MAP, the Q function is:

$$Q((\boldsymbol{p}, \boldsymbol{\pi}), (\overline{\boldsymbol{p}}, \overline{\boldsymbol{\pi}})) = \mathbb{E}_{\{\boldsymbol{y}, \boldsymbol{z} \mid \underline{R}_i, \boldsymbol{p}, \boldsymbol{\pi}\}} \mathcal{L}_C(\boldsymbol{p}, \boldsymbol{\omega}, \boldsymbol{y}, \boldsymbol{z}) + \log f_0(\boldsymbol{p}, \boldsymbol{\omega})$$

where $\mathcal{L}_C$ is the complete-data log likelihood function, and f_0 is the function that represents the priors of the prior distributions for $\boldsymbol{\pi}$ and $\boldsymbol{p}$. Given these priors are formulated independently of each other, the total contribution of the priors can be written as

$$\begin{aligned}
\log(f_0(\boldsymbol{p}, \boldsymbol{\pi})) &= \log\left(f_0(\boldsymbol{p}) \cdot f_0(\boldsymbol{\pi})\right) \\
&= \log(f_0(\boldsymbol{p})) + \log(f_0(\boldsymbol{\omega}))
\end{aligned}$$

M-step: Updated support parameters $\boldsymbol{p}$:

The likelihood of ranking R_i for judge i and the latent variables y_{it}, conditional on the previous estimates for the support parameters is:

$$L(R_i, \boldsymbol{y} \mid \overline{\boldsymbol{p}}) \propto \prod_{i=1}^{M} \prod_{t=1}^{n_i} [\overline{p}_{r(i,t)} \exp(-\lambda_{it} y_{it})]$$

taking logarithms gives:

$$\begin{aligned}
\log(L(R_i, \boldsymbol{y} \mid \overline{\boldsymbol{p}})) = \mathcal{L}_C(R_i, \boldsymbol{y}) &= \sum_{i=1}^{M} \sum_{t=1}^{n_i} [\log \overline{p}_{r(i,t)} - \lambda_{it} y_{it}] \\
&= \sum_{i=1}^{M} \sum_{t=1}^{n_i} [\log \overline{p}_{r(i,t)} - (\sum_{j=1}^{N} \overline{p}_j - \sum_{q=1}^{t-1} \overline{p}_q) y_{it}]
\end{aligned}$$

up to a constant. Note that:

- At preference level t for judge i, the term

$$\sum_{j=1}^{N} \overline{p}_j - \sum_{q=1}^{t-1} \overline{p}_q$$

can be reimagined as the sum of the support parameters for the items that have not yet been chosen i.e. :

$$\sum_{j=1}^{N} \delta_{itj} p_j,$$

where $\delta_{itj} = 1$ if item j is still available at preference level t and 0 otherwise.

- The product over the ranking positions can be rewritten in terms of an indicator variable. Specifically,

$$\prod_{t=1}^{n_i} \overline{p}_{r(i,t)} = \prod_{t=1}^{n_i} \prod_{j=1}^{K} p_j^{\mathbb{1}\{r(i,t)=j\}} = \prod_{j=1}^{K} p_j^{\sum_{t=1}^{n_i} \mathbb{1}\{r(i,t)=j\}} = \prod_{j=1}^{K} p_j^{u_{ij}},$$

where

$$u_{ij} = \sum_{t=1}^{n_i} \mathbb{1}\{r(i,t) = j\}$$

is the number of times item j appears (i.e. is selected) in judge i's ranking.

Thus, combining over all judges $i = 1, \ldots, M$, the complete-data log-likelihood for the support parameters can be written as

$$\mathcal{L}_C(\boldsymbol{R}, \boldsymbol{y} \mid \boldsymbol{p}) = \sum_{k=1}^{K} \sum_{i=1}^{M} \left\{ \sum_{j=1}^{N} u_{ij} \log p_{jk} - \sum_{t=1}^{n_i} \left[\left(\sum_{j=1}^{N} \delta_{itj} \, p_{jk} \right) y_{it} \right] \right\}.$$

up to a constant. For Bayesian inference, a gamma prior is included for the support parameters p_{jk}:

$$\mathbb{P}(p_{jk}) \propto p_{jk}^{c_{jk}-1} \exp(-d_g, p_{jk}),$$

Thus, the full complete-data log-posterior for p_{jk} becomes

$$\mathcal{L}(\boldsymbol{p}) = \sum_{k=1}^{K} \left[\left\{ \underbrace{\sum_{i=1}^{M} z_{ik} u_{ij}}_{\gamma_{jk}} + c_{jk} - 1 \right\} \log p_{jk} - \left(\sum_{i=1}^{M} z_{ik} \sum_{t=1}^{n_i} \delta_{itj} y_{it} \right) p_{jk} + (c_{jk} - 1) \log p_{jk} - d_k \, p_{jk} \right]$$

Up to a constant. To obtain the updated support parameter estimate the complete-data log likelihood function is differentiated with respect to p_{jk}:

$$\frac{\partial \mathcal{L}(\boldsymbol{p})}{\partial p_{jk}} = \frac{\gamma_{jk} + c_{jk} - 1}{p_{jk}} - \left(d_k + \sum_{i=1}^{M} z_{ik} \sum_{t=1}^{n_i} \delta_{itj} \, y_{it} \right).$$

Equating the derivative equal to zero yields:

$$\frac{\gamma_{jk} + c_{jk} - 1}{p_{jk}} = d_k + \sum_{i=1}^{M} z_{ik} \sum_{t=1}^{n_i} \delta_{itj} \, y_{it},$$

$$\implies$$

$$p_{jk} = \frac{\gamma_{jk} + c_{jk} - 1}{d_g + \displaystyle\sum_{i=1}^{M} z_{ik} \sum_{t=1}^{n_i} \delta_{itj} y_{it}},$$

where z_{ik} is the latent indicator that judge i belongs to component g.

M-step: Updated mixture weights $\boldsymbol{\omega}$:

For the mixing weights $\boldsymbol{\pi}$, the complete–data likelihood contribution is

$$\log L(\boldsymbol{\pi}) = \mathcal{L}(\boldsymbol{\pi}) = \sum_{i=1}^{M} \sum_{k=1}^{K} z_{ik} \log \omega_g.$$

Assuming a Dirichlet prior for $\boldsymbol{\pi}$:

$$\mathbb{P}(\boldsymbol{\pi}) \propto \prod_{k=1}^{K} \omega_k^{\alpha_k - 1},$$

so that

$$\log \mathbb{P}(\boldsymbol{\pi}) = \sum_{k=1}^{K} (\alpha_k - 1) \log \omega_k$$

Thus, up to a constant, the complete–data log–posterior is

$$Q(\boldsymbol{\pi}) = \sum_{i=1}^{M} \sum_{k=1}^{K} z_{ik} \log \pi_k + \sum_{k=1}^{K} (\alpha_k - 1) \log \pi_k.$$

$Q(\boldsymbol{\pi})$ can be differentiated subject to the constraint that the mixing weights must sum to 1, i.e. :

$$\sum_{k=1}^{K} \pi_k = 1$$

A Lagrange multiplier λ is introduced with respect to this constraint, and a Lagrangian function is defined:

$$\mathcal{L}(\boldsymbol{\pi} \mid \lambda) = \sum_{k=1}^{K} \left[\left(\sum_{i=1}^{M} z_{ik} + \alpha_k - 1 \right) \log \pi_k \right] + \lambda \left(1 - \sum_{k=1}^{K} \pi_k \right)$$

To maximise, the derivative is found with respect to π_k and is set to equal 0:

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi} \mid \lambda)}{\partial \pi_k} = \left(\sum_{i=1}^{M} z_{ik} + \alpha_k - 1 \right) \times \left(\frac{1}{\pi_k} \right) - \lambda$$

$$0 = \frac{\sum_{i=1}^{M} z_{ik} + \alpha_k - 1}{\pi_k} - \lambda$$

Therefore,

$$\hat{\pi}_k = \frac{\sum_{i=1}^{M} z_{ik} + \alpha_k - 1}{\lambda}$$

$$\sum_{k=1}^{K} \pi_k = \frac{1}{\lambda} \sum_{k=1}^{K} \sum_{i=1}^{M} (z_{ik} + \alpha_k - 1) = 1$$

$$\implies$$

$$\lambda = \sum_{k=1}^{K} \sum_{i=1}^{M} (z_{ik} + \alpha_k - 1)$$

After substituting λ,

$$\hat{\pi}_k = \frac{\sum_{i=1}^{M} z_{ik} + \alpha_k - 1}{\lambda}$$

$$= \frac{\sum_{i=1}^{M} z_{ik} + \alpha_k - 1}{\sum_{k'=1}^{K} \sum_{i=1}^{M} (z_{ik'} + \alpha_{k'} - 1)}$$

$$= \frac{\sum_{i=1}^{M} z_{ik} + \alpha_k - 1}{\sum_{k'=1}^{K} z_{ik'} + M - K}$$

7.3 Gibbs Sampling

In this model, a Gibbs Sampling procedure is used to ascertain the uncertainty of the final estimates of the parameters, namely the support parameters $\boldsymbol{p}$ and the mixing weights $\boldsymbol{\pi}$. This model has other parameters, the latent mixture allocation variables $\boldsymbol{z}$ and the latent exponential data augmentation variables $\boldsymbol{y}$, which also need to be drawn from their individual distributions.

- **Sampling the Support Parameters $\boldsymbol{p}$:**

  Recall that the Plackett-Luce support parameters p_{jk} indicate the support for item j in component k. These parameters influence the complete-data log-likelihood through three distinct contributions: (1) the Plackett–Luce ranking likelihood, (2) the exponential terms introduced via data augmentation, and (3) the log-prior arising from the Gamma distribution assumed for each p_{jk}.

1. For each judge i, item $j's$ support parameter for mixing component k is only influenced if the judge is a member of that component and if they have revealed a preference for that item. Hence, the ranking likelihood contributes over the judges to the sampling through $\sum_{i=1}^{M} z_{ik} u_{ij}$.

2. At each preference level t, judge i has a latent exponential variable y_{it}. At each stage the support parameter p_{jk} is only affected if the judge ranks it, and if the judge is a member of component k. The equation $A_{ik} = \sum_{i=1}^{N} z_{ik} \sum_{t=1}^{n_i} \delta_{ijt} y_{it}$ is defined efficiently count such scenarios.

3. The prior distribution of p_{jk} assumes that it is a gamma distributed random variable:

$$\mathbb{P}(p_{jk}) \propto p_{jk}^{c_{jk}-1} \exp(-d_k p_{jk})$$

Combining all three of these contributions, the support parameters can be sampled as conditionally Gamma random variables:

$$\mathbb{P}(p_{jk}) \propto p_{jk}^{\left(\sum_{i=1}^{M} z_{ik} u_{ij} + c_{jk} - 1\right)} \exp\left[-p_{jk}\left(d_k + A_{ik}\right)\right].$$

$\implies$

$$p_{jk} \mid \sim \text{Gamma}\left(\sum_{i=1}^{M} z_{ik} u_{ij} + c_{jk}, d_k + A_{ik}\right)$$

- **Sampling the Mixture Weights $\boldsymbol{\pi}$:**

Assuming a Dirichlet prior, the mixture weights $\boldsymbol{\pi}$ can be sampled from a conditional Dirichlet distribution. This is a direct result of the likelihood of the mixture weights have a multinomial structure, as a result of the latent component allocation variables $\boldsymbol{z}$:

$$\mathbb{P}(\boldsymbol{\pi} \mid \boldsymbol{z}) \propto \mathbb{P}(\boldsymbol{z} \mid \boldsymbol{\pi})\mathbb{P}(\boldsymbol{\pi})$$
$$= \prod_{i=1}^{M} \prod_{k=1}^{K} \pi_k^{z_{ik}} \times \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}$$

($b_k = \sum_{i=1}^{M} z_{ik}$ is introduced for notational convenience)

$$= \prod_{k=1}^{K} \pi_k^{b_k + \alpha_k - 1}$$

The last part of the derivation shows us that $\boldsymbol{\pi}$ is conditionally Dirichlet distributed:

$$\boldsymbol{\pi} \mid \boldsymbol{z} \sim \text{Dirichlet}(b_1 + \alpha_1, b_2 + \alpha_2, \ldots, b_K + \alpha_K)$$

- **Sampling the Exponential Latent Variables $\boldsymbol{y}$:**

In the Partial Plackett-Luce density, the term $\sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q$ is replaced by the exponential random variable y_{it}, where:

$$y_{it} \sim \mathcal{E}\left(\sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q\right)$$

for judge i at preference level t. Thus, y_{it} is also sampled using this exponential augmentation during Gibbs Sampling.

- **Sampling the Mixture Indicator Variables z:**

  In the construction of the model, it was assumed that the mixture indicator variables for the ith judge are distributed, conditional on the latest estimates for the mixture weights, by a multinomial case i.e. :

  $$(z_{i1}, \ldots, z_{ik}, \ldots, z_{iK}) \sim \text{Multinomial}(1, \boldsymbol{\pi})$$
  $$\sim \text{Multinomial}(1, \hat{\pi}_1, \ldots, \hat{\pi}_k, \ldots, \hat{\pi}_K)$$

  where

  $$\hat{\pi}_k = \frac{\pi_k \mathbb{P}_{PPL}(R_i \mid \boldsymbol{p}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathbb{P}_{PPL}(R_i \mid \boldsymbol{p}_{k'})}$$

By cycling through these steps, samples from the joint posterior distribution of $\boldsymbol{p}, \boldsymbol{\omega}, \boldsymbol{z}$, and $\boldsymbol{y}$ are produced. These samples can then be used for Bayesian inference and to quantify how well the model is capturing any heterogeneity in the data. Note that in practice, label switching issues can be inherent in mixture models. Hence, well calibrated relabeling algorithms are utilised in order to post-process the MCMC samples produced [31].

In their paper, Mollica and Tardella fit their data and parameters to various model selection metrics to infer which criterion would be best for a partially ranked data model. Specifically, they fit different formulations of the Deviance Information Criterion (DIC), the Bayesian Information Criterion–Monte Carlo (BICM), and the Bayesian Predictive Information Criterion (BPIC) to various scenarios. They fitted Plackett-Luce mixtures with various K components under a variety of data sets. Specifically, they used simulated data sets that induced various censored data settings to evaluate the criterions performance under various resertions. They found that their first formulation of the Deviance Information Criterion (DIC_1) most reliably recovers the true number of components in challenging partial ranked data scenarios. In contrast, the different forms of the BICM criteria were found to inflexibly under fit as the amount of complexity increases i.e. as the amount of partial rankings increases.

The Deviance Information Criterion uses Deviance, a goodness of fit statistic, which is penalised by the effective number of parameters p_D in a model. Formally, letting:

$$D(\theta) = -2 \log L(\theta), \quad \bar{D} = \mathbb{E}[D(\theta) \mid R_i], \quad \hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \mathbb{P}(\theta \mid R_i)$$

with the effective number of parameters $p_D = \overline{D} - D(\hat{\theta}_{\text{MAP}})$, then:

$$DIC_1 = \overline{D} + p_D$$

8 A Mixture of Experts Model for Rank Data

8.1 Background

Mixture of Experts (MoE) models have been widely applied across various disciplines. In their 2008 paper, Gormley and Murphy adapted the MoE framework to develop an algorithmic clustering model for rank data based on a probabilistic interpretation of the framework [2]. By incorporating covariates—which in an electoral setting can range from broader social factors such as government satisfaction levels and socioeconomic status to personal factors such as age and gender—this model provides stronger insights into the underlying composition of the judges revealing their preferences. Such insights are not achievable in a simple mixed model, as the model assumes that the mixture weights are the same for every judge and cannot flexibly relate group-membership probabilities to observed predictors.

In contrast to other tree-based clustering models, such as Classification trees, the Mixture of Experts Model for ranked data provides better insights into heterogeneity in the data. The gating network assigns each judge i with some probability π_{ik} of being a member of expert network k. The expert networks in a Mixture of Experts Model for Rank Data typically each explain the specific effect of one particular covariate or a small subset of all of the covariates. That said, it is often the case that judges preferences are influenced by a combination of such factors, hence a soft probabilistic soft clustering algorithm is preferable. In a tree-based clustering model, the input space is split on a hard rules based system. Hence, each data point, or judge in this case, can only be assigned to one cluster. An analogous argument can be made in support of a MoE model over a regular Mixture Model.

8.2 Model Specification

Formally, the conditional probability of observing judge i's ranking R_i, given their covariates $\underline{x}_i$, is:

$$\mathbb{P}(\underline{R}_i|\underline{x}_i) = \sum_{k=1}^{K} \pi_{ik}\mathbb{P}_{\mathrm{B}}(\underline{R}_i|\theta_k)$$

where $\mathbb{P}_{\mathrm{B}}$ is the Benter model for rank data's likelihood function, as discussed in Section 2, and with θ_k representing the parameters of expert network k. The parameters include the gating network parameters β_{kl}, which represent the coefficient of covariate l in the k'th expert network's logistic regression function, and the Benter support parameters $\underline{p}_k$, which represent the support for each item in the kth expert.

The Benter dampening parameters $\boldsymbol{\alpha}$ are global estimates that are not specific to a particular expert. They represent the certainty with which the judges select particular items at preference level t. There is evidence in the literature that explicitly modeling the phenomena captured by dampening parameters is crucial for accurately uncovering stage-wise selection effects in rank data [32].

8.2.1 Gating Network Coefficients and Parameters

- The gating network parameters β_{kl} represent the coefficient of covariate l in the k'th expert network's logistic regression function.

- The gating network coefficients for the K expert networks $\underline{\pi}_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik}, \ldots, \pi_{iK})$ for judge i are modeled using their L coefficients $\underline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{il}, \ldots, x_{iL})$ via:

$$\log\left(\frac{\pi_{ik}}{\pi_{i1}}\right) = \beta_{k1}x_{i1} + \beta_{k2}\omega_{i2} + \cdots + \beta_{kl}x_{il} + \cdots + \beta_{kL}x_{iL}$$

where expert network 1 acts as the baseline network, and β_{k0} acts as the intercept term. For ease of interpretation and inference, each covariate parameter β_{kl} are either a binary variable, or normalised to attain a value between 0 and 1.

8.3 Parameter Estimation

The foundation for maximum likelihood expectation in this Mixture of Experts for Rank Data model is a hybrid Expectation-Minorization-Maximisation (EMM) approach, which combines the concepts of the Expectation-Maximisation (EM) and Minorization-Maximisation (MM) algorithms to allow for the derivation of a closed form for the maximised parameters.

The likelihood function here is the product of each density of each observation:

$$
\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{R}, \boldsymbol{x}) = \mathcal{L}(\boldsymbol{\beta}, \boldsymbol{p}, \underline{\alpha}|\boldsymbol{x}, \boldsymbol{\omega}) = \prod_{i=1}^{M} \mathbb{P}(\underline{R}_i|\underline{x}_i, \boldsymbol{\beta}, \boldsymbol{p}, \boldsymbol{\alpha})
$$

$$
= \prod_{i=1}^{M} \sum_{k=1}^{K} \pi_{ik}(\underline{x}_i) \mathbb{P}_B(\underline{R}_i|\underline{p}_k, \boldsymbol{\alpha})
$$

The enumerative nature of this formula makes direct attaining of MLEs difficult. Hence, a complete data likelihood function is derived, based on latent variables z_{ik} with:

$$
z_{ik} = \begin{cases} 1 & \text{voter } i \text{ is a complete member of expert } k \\ 0 & \text{otherwise} \end{cases}
$$

where:

$$
\mathcal{L}_C(\boldsymbol{\beta}, \boldsymbol{p}, \underline{\alpha}|\boldsymbol{x}, \boldsymbol{\omega}, \boldsymbol{z}) = \prod_{i=1}^{M} \prod_{k=1}^{K} [\pi_{ik}(w_i) \mathbb{P}_B(x_i|p_k, \alpha)]^{z_{ik}}
$$

Consequentially, the log-likelihood of the complete data likelihood function is:

$$
\log(\mathcal{L}_C) = \log \prod_{i=1}^{M} \prod_{k=1}^{K} \{\pi_{ik}(\underline{x}_i) \mathbb{P}_B(\underline{R}_i|\underline{p}_k, \boldsymbol{\alpha}))\}^{z_{ik}}
$$

$$
= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} [\log(\pi_{ik}(\underline{x}_i)) + \log \mathbb{P}(\underline{R}_i|\underline{p}_k, \boldsymbol{\alpha})]
$$

$$
= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log \left(\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i)} \right) + \log \left(\sum_{t=1}^{n_i} \frac{p_{kc(i,t)}^{\alpha_t}}{\sum_{s=t}^{N} p_{kc(i,s)}^{\alpha_t}} \right) \right]
$$

which implies that the Q function has the form:

$$
Q = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log(\exp(\underline{\beta}_k^T \underline{x}_i)) - \log \left(\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i) \right) + \sum_{t=1}^{n_i} \log \left(\frac{p_{kc(i,t)}^{\alpha_t}}{\sum_{s=t}^{N} p_{kc(i,s)}^{\alpha_t}} \right) \right]
$$

$$
= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log(\exp(\underline{\beta}_k^T \underline{x}_i)) - \log \left(\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i) \right) + \sum_{t=1}^{n_i} \alpha_t \log p_{kc(i,t)} - \log \sum_{s=t}^{N} p_{kc(i,s)}^{\alpha_t} \right]
$$

There are still a few issues with this function. The parameters that we are estimating are coupled to each other i.e. they are not split into separate terms. Maximisation is easier when they are uncoupled from each other. Also, the expression is still combinatorial in nature and an analytical derivative for maximisation cannot be found as a result. Hence, a Minorization-Maximisation algorithm is used in the M step of the Expectation-Maximisation algorithm to estimate the parameters $\boldsymbol{\beta}, \boldsymbol{p}$, & .

8.4 Expectation-Minorization-Maximization Algorithm

The estimation is based on the Q-function, which is the log-likelihood of the complete data likelihood function. The logarithm is taken to ensure that the derivative is easier to calculate as it can be represented as a sum. The monotone transformation ensures this, while also preserving the location of the extreme points. There are four steps to this hybrid EMM model:

1. **Choose initial estimates of the parameters:**

   An initial estimate of each of the parameters $\boldsymbol{\beta}^{(0)}, \boldsymbol{p}^{(0)}$, and $\boldsymbol{\alpha}^{(0)}$. is calculated, using 500 iterations of a simple mixture model. This ensures reasonable convergence to the true parameters, as the Expectation-Maximisation model is only locally convergent.

2. **E step: Compute the estimates for the latent variables $\hat{z}_{ik}$:**

   Given the current estimates for the parameters $\boldsymbol{\beta}^{(h)}, \boldsymbol{p}^{(h)}$, & $\boldsymbol{\alpha}^{(h)}$, estimates for the $\hat{z}_{ik}$ are derived using the formula:

   $$\hat{z}_{ik} = \frac{\mathbb{P}(\underline{R}_i|\underline{p}_k^{(h)}, \alpha_t^{(h)})\pi_{ik}^{(h)}}{\sum_{k'=1}^{K} \mathbb{P}(\underline{R}_i|\underline{p}_{k'}^{(h)}, \alpha_t^{(h)})\pi_{ik'}^{(h)}}$$

3. **M step: Maximise a surrogate "Q - Function" after substituting the values obtained for $\hat{z}_{ik}$ into the complete data log likelihood function:**

   The Q-function is the expected complete data log-likelihood. In this step, we find the new parameter estimates $\boldsymbol{\beta}^{(h+1)}, \boldsymbol{p}^{(h+1)}$, & $\boldsymbol{\alpha}^{(h+1)}$ using:

   $$Q = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\exp(\underline{\beta}_k^T \underline{x}_i) - \log\left(\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i) \right) + \sum_{t=1}^{n_i} \alpha_t \log p_{kc(i,t)} - \log \sum_{s=t}^{N} p_{kc(i,s)}^{\alpha_t} \right]$$

   Each parameter is maximised individually, conditional on the values of other remaining parameters, which are held constant. This section draws upon various sources to explain the methods used in the Maximisation step of the Mixture of Experts for Rank Data model [33, 34, 35]. Given there is a set of parameters that are being estimated, a conditional maximisation can be induced [36], allowing for $\boldsymbol{\alpha}^{(h+1)}$ to be inferred individually conditional on the values $\boldsymbol{\beta}^{(h+1)}$ & $\boldsymbol{p}^{(h+1)}$, and so on. This algorithm, alongside both of the inequalities derived previously, provide a useful basis in the construction of surrogate functions and their maximisation. The derivations for the maximisations are provided in Appendix A.

   - The latest estimates for the logistic regression function parameters $\boldsymbol{\beta}^{(h)}$ and the Benter dampening parameters $\boldsymbol{\alpha}^{(h)}$ are substituted into the following expression to achieve the estimates for the Benter support parameters for the $h + 1$st iteration:

   $$\hat{p}_{jk}^{(h+1)} = \frac{\omega_{jk}}{\sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N+1} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \delta_{ijs} \right)}$$

For ease of notation, ω_{kj} is defined:

$$\omega_{jk} = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \overline{\alpha}_t \mathbb{1}_{j=c(i,t)}$$

and the indicator functions 1 and δ are defined such that:

$$\mathbb{1}_{j=c(i,s)} = \begin{cases} 1 & \text{voter } i \text{ gives candidate } j \text{ their } s \text{ preference level} \\ 0 & \text{otherwise} \end{cases}$$

&

$$\delta_{ijs} = \begin{cases} 1 & \text{if } j = r_{(i,s)} \text{ for } 1 \le s \le n_i \\ 1 & \text{if } j \ne r_{(i,l)} \text{ for } 1 \le l \le n_i, \text{ but } s = N+1 \\ 0 & \text{otherwise} \end{cases}$$

- For the $h+1$st iteration, the Benter dampening parameter is updated, conditional on the estimates of the previous iteration, via the following expression:

$$\hat{\alpha}_t = \frac{\sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log \hat{p}_{kc(i,s)} + \left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} -(\log \hat{p}_{kc(i,s)}) \hat{p}_{kc(i,s)}^{\overline{\alpha}} + \overline{\alpha}_t (\log \hat{p}_{kc(i,s)})^2 \right) \right] \cdot \mathbb{1}_{t \le n_i}}{\sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} (\log \hat{p}_{kc(i,s)})^2 \right) \right] \cdot \mathbb{1}_{t \le n_i}}$$

where the indicator function $\mathbb{1}_{t \le n_i}$ is defined such that:

$$\mathbb{1}_{t \le n_i} = \begin{cases} 1 & \text{voter } i \text{ expresses a preference for candidate } t \\ 0 & \text{otherwise} \end{cases}$$

- The multinomial logistic regression functions are updated using the following update step:

$$\underline{\beta}_k^{(h+1)} = \underline{\beta}_k^{(h)} - \boldsymbol{B}^{-1} q'(\underline{\beta}_k^{(h)})$$

For $\boldsymbol{B} = -\frac{1}{4} \sum_{i=1}^{M} \underline{x}_i \underline{x}_i^T$.

4. **Convergence Check:** If the estimates have converged, then stop the process. If not, then return to Step 2.

The optimal amount of expert networks K is inferred using the Bayesian Information Criterion (BIC) [37]:

$$BIC = 2(\text{maximised likelihood}) - (\text{number of parameters}) \times \log(\text{number of judges})$$

The larger the value of the Criterion, the better the fitting of the particular model being considered. BIC's balance between goodness of fit and model complexity makes it well suited in determining how many expert networks best describe the heterogeneity in the observed data.

9 A Bayesian Mixture of Experts Model for Partially Ranked Data

Both the Mollica and Tardella and the Gormley and Murphy Models prove to be stable and reliable avenues for inference, as demonstrated both in their original work and in the results presented in this project. Nevertheless, both models have their respective limitations. The Gormley and Murphy model is in practice only workable on full ranking or nearly full ranking data. In the dataset they employ from a 1997 Irish Presidential Election opinion poll, respondents were asked to provide up to five rankings for a race with five candidates. The algorithm provided in the accompanying C code deterministically fills in unrevealed rankings rather than handling them probabilistically or randomly. For instance, if only three candidates are ranked, say $r(1) = B, r(2) = D$, and $r(3) = E$, the Gormley and Murphy model proceeds through the candidate list from top to bottom (starting with A, then B, and so on) and fills in the missing items accordingly. Hence, the algorithm is biased towards whichever candidate happens to be first in the candidate list i.e. first in alphabetical order. Although this deterministic process introduces minimal bias in the 1997 election data, since a substantial percentage of respondents revealed at least three rankings, such an approach is likely to be problematic in elections with more candidates or in less comprehensive surveys (for example, the Ireland Thinks 2020 Irish General Election survey, which collected only the top four preferences from voters among eleven parties).

In contrast, the Mollica and Tardella model explicitly accounts for partial rankings and fully quantifies uncertainty through the Bayesian Gibbs Sampling. That said, it does not provide a model based framework to explain why individual observations are assigned to specific clusters. Instead, it offers a global mixing weight for each cluster rather than a convex combination of mixing weights for each observation over the K clusters.

Hence, this section will introduce a model that combines the explanatory power of the Gormley and Murphy framework with the partial ranking and Bayesian structure of the Mollica and Tardella model. In the current literature, Bayesian Mixture of Experts models have not been applied to partially ranked data. Accordingly, suitable data augmentation structures will be established to enable this novel approach. There is no obvious way to model the effect of partial rankings on the Benter dampening parameters, so they will be excluded from this model.

9.1 Model Overview

In a Mixture of Experts for Partially Ranked Data setting, the posterior probability $\mathbb{P}(\underline{R}_i \mid \underline{x}_i)$ of voter i's ranking $\underline{R}_i$ (where voter i has covariates $\underline{x}_i$) is given by:

$$\mathbb{P}(\underline{R}_i \mid \underline{x}_i) = \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\pi_{ik} \prod_{l=1}^{n_i} \mathbb{P}_{PPL}(\underline{R}_i \mid \boldsymbol{p}) \right]$$

$$= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i)} \times \prod_{l=1}^{n_i} \frac{p_{r(i,l,k)}}{\sum_{j=1}^{J} p_{r(i,j,k)} - \sum_{q=1}^{l-1} p_{r(i,q,k)}} \right]$$

where $p_{r(i,l,k)}$ is the support parameter in the k'th cluster for the candidate that voter i ranked in the l'th position. To allow for closed form updates in the EM algorithm, this equation is augmented with a latent variable z_{ik}:

$$\mathbb{P}(\underline{R}_i \mid \underline{x}_i) = \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\left\{ \left(\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i)} \right) \times \prod_{l=1}^{n_i} \left(\frac{p_{r(i,l,k)}}{\sum_{j=1}^{J} p_{r(i,j,k)} - \sum_{q=1}^{l-1} p_{r(i,q,k)}} \right) \right\} \right]^{z_{ik}}$$

As discussed in the previous models that were introduced, both of the denominators of in this equation are problematic due to the coupling of the parameters across the various clusters. Hence, Data Augmentation methods will also be necessary in this model. So far in this report, the exponential augmentation for the Partial Plackett-Luce density is well defined. For a Mixture of Experts model, the Pólya-Gamma augmentation is still appropriate, but algebraic tricks are necessary to achieve a closed form, as a consequence of the usual approaches in the literature being for logistic models, rather than the multinomial form induced by the Mixture of Experts framework. This novel approach offers a flexible and probabilistically sound method for clustering partially ranked data, resulting in each observation being assigned a convex combination of mixing weights over the K clusters, rather than a single global assignment, thereby addressing limitations present in previous models.

9.2 Latent Variable Augmentation

9.2.1 Exponential Augmentation

Given the logical link between the 'exponential race' concept and that of ranked data, as described in REF, an augmentation grounded in this framework will again be implemented in this Bayesian Mixture of Experts for Partially Ranked Data Model. The Partial Plackett-Luce density that is used in this model is defined as:

$$\mathbb{P}(\underline{R}_i \mid \underline{x}_i) = \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\left\{ \prod_{l=1}^{n_i} \left(\frac{p_{r(i,l,k)}}{\sum_{j=1}^{J} p_{r(i,j,k)} - \sum_{q=1}^{l-1} p_{r(i,q,k)}} \right) \right\} \right]^{z_{ik}}$$

for M voters, who each reveal n_i preferences, with the underlying assumption that there are K clusters or experts. Also, $p_{r(i,l,k)}$ represents the support parameter in expert k for the candidate that voter i ranks in the l'th position Hence, a suitable latent variable y_{it} is defined, such that:

$$y_{it} \sim \exp\left(\sum_{j=1}^{J} p_{r(i,j,k)} - \sum_{q=1}^{l-1} p_{r(i,q,k)} \right)$$

9.2.2 Pólya-Gamma Augmentation

Recall from THM HYP Theorem 6.3 that, for $\omega \sim \mathrm{PG}(b,0)$, for $b > 0$:

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = 2^{-b} e^{\kappa \psi} \int_0^\infty e^{-\omega \psi^2 / 2} p(\omega) d\omega$$

which naturally leads to an augmented form for the logit link function $\frac{e^\psi}{1+e^\psi}$, for $\psi = \boldsymbol{\beta x}^T$. That said, the Mixture of Experts mixing weight likelihood has a slightly different form:

$$\pi(\underline{x}_i)_k = \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k'=1}^{K} \exp(\underline{\beta}_{k'}^T \underline{x}_i)} \right]$$

Hence, a 'stick-breaking' routine will be utilised to induce a closed form for updates in inference steps [28]. For voter i, the softmax probability for assigning them to expert k is given by:

$$\psi_{(ik)} = \beta_k^T x_i - c_i$$

where $\quad c_i = \log\left(\sum_{l \neq k} \exp(\beta_l^T x_i)\right)$. This new representation casts the multinomial likelihood into a form analogous to the standard logistic link function:

$$
\begin{aligned}
\pi(\underline{x}_i)_k &= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k'=1}^{K} \exp(\underline{\beta}_{k'}^T \underline{x}_i)} \right] \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\frac{\exp\left(\boldsymbol{\beta}_k^T \mathbf{x}_i - \log \sum_{\ell=1}^{K} \exp\left(\boldsymbol{\beta}_\ell^T \mathbf{x}_i\right)\right)}{1 + \exp\left(\boldsymbol{\beta}_k^T \mathbf{x}_i - \log \sum_{\ell=1}^{K} \exp\left(\boldsymbol{\beta}_\ell^T \mathbf{x}_i\right)\right)} \right] \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\frac{\exp(\psi_{ik})}{1 + \exp(\psi_{ik})} \right]
\end{aligned}
$$

These reformulations facilitate the derivation of closed-form updates for $\boldsymbol{p}$ and $\boldsymbol{\beta}$ during the estimation processes, a MAP estimation procedure based on the EM algorithm, and a full Bayesian inference scheme implemented via Gibbs sampling.

9.3 Maximum a Posteriori Estimation

The maximum value (mode) of the posterior distribution will be attained using MAP approximation. This will be induced by an ECM algorithm, where each parameter is maximised, holding the latest estimates and the latest estimates for the previously maximised parameters in the same iteration as constants. The ECM algorithm is particularly helpful in the maximisation of the Pólya-Gamma augmented density contribution to the complete data, due to the 'stick breaking' form that is used to prevent coupling. The unaugmented complete data likelihood function is:

$$
\begin{aligned}
\mathbb{P}(\underline{R}_i \mid \underline{x}_i) &= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\pi_{ik} \prod_{l=1}^{n_i} \mathbb{P}_{PPL}(\underline{R}_i \mid \boldsymbol{p}) \right]^{z_{ik}} \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \left[\left\{ \left(\frac{\exp(\underline{\beta}_k^T \underline{x}_i)}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i)} \right) \times \prod_{l=1}^{n_i} \left(\frac{p_{r(i,l,k)}}{\sum_{j=1}^{J} p_{r(i,j,k)} - \sum_{q=1}^{l-1} p_{r(i,q,k)}} \right) \right\} \right]^{z_{ik}}
\end{aligned}
$$

The *ECM* algorithm has two steps, the E-step and the M-step. In this model, there are now three latent variables - the typical mixture assignment variable ($\boldsymbol{z}$), the exponential augmentation variable ($\boldsymbol{y}$), and the Pólya-Gamma augmentation variable ($\boldsymbol{\omega}$). At the E-step of each iteration, the expected value of each of these latent variables is calculated given the latest estimates for the models parameters. In this model, only the mixture assignment and the Pólya-Gamma latent variables will be explicitly calculated, due to the simpler representation of the expected value of an exponentially distributed random variable. The expected values are:

$$\eta_{ik} = \mathbb{E}[z_{ik}] = \frac{\pi_k(\underline{x}_i)\, \mathbb{P}_{\text{PPL}}(R_i \mid p_k)}{\sum_{k'=1}^{K} \pi_{k'}(\underline{x}_i) \mathbb{P}_{\text{PPL}}(R_i \mid p_{k'})}$$

$$\chi_{jk} = \mathbb{E}[\beta_{jk}] = \frac{1}{2\psi_{ik}} \tanh\left(\frac{\psi_{ik}}{2}\right)$$

Now, the M-step will be laid out for both of the parameters that are being estimated. In MAP estimated, the Q-function also includes the prior distribution of each parameter. Hence, the Q-

function takes the form

$$Q((\boldsymbol{p}, \boldsymbol{\omega})(\overline{\mathbf{p}}, \overline{\boldsymbol{\omega}})) = \mathbb{E}_{\{\boldsymbol{\omega}, \boldsymbol{y}, \boldsymbol{z} | \boldsymbol{R}, \boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{p}\}}[\log \mathcal{L}_C(\{\boldsymbol{\beta}, \boldsymbol{p}, \boldsymbol{\omega}, \boldsymbol{y}, \boldsymbol{z}\}] + \log f_0(\boldsymbol{\beta}, \boldsymbol{p})$$

$$= \sum_{i=1}^{M} \sum_{k=1}^{K} \eta_{ik} \left[\kappa_{ik} \, \psi_{ik} - \frac{1}{2} \, \chi_{ik} \, \psi_{ik}^2 \right]$$

$$+ \sum_{i=1}^{M} \sum_{k=1}^{K} \eta_{ik} \sum_{t=1}^{n_i} \log p_{k,r(i,t)} \; - \; \sum_{i=1}^{M} \sum_{k=1}^{K} \eta_{ik} \sum_{t=1}^{n_i} u_{itk} \, \lambda_{itk}$$

9.3.1   M-step - Updating the Support Parameters

For the Q-function of this model, when differentiating with respect to the support parameters $\boldsymbol{p}$, the procedure is the same as that seen in the Mollica and Tardella mixture model, which is described in Section 7. Therefore, the updated support parameters for the M-step can be represented as:

$$\hat{p}_{jk} = \frac{c_{jk} - 1 + \hat{\gamma}_{jk}}{d_k + \sum_{i=1}^{M} \eta_{ik} \sum_{t=1}^{n_i} \frac{\delta_{itj}}{\sum_{j=1}^{N} \delta_{itj}}}$$

- where γ_{jk} is the effective count i.e. the number of times p_{jk} appears in the normalisation across the data, or more formally:

$$\gamma_{ik} = \sum_{i=1}^{M} z_{ik} u_{ij}$$

with:

$$u_{ij} = \begin{cases} 1 & \text{if } j \in R_i \\ 0 & \text{otherwise} \end{cases}$$

- δ_{itj} is defined to account for the edge case where the voter writes down all but one preference, which in practice is a full ranking but would not be recognised as one in the status quo formulation:

$$\delta_{itj} = \begin{cases} 1 & \text{if } j = r_{(i,t)} \text{ for } 1 \leq t \leq n_i \\ 1 & \text{if } j \neq r_{(i,l)} \text{ for } 1 \leq l \leq n_i, \text{ but } t = N + 1 \\ 0 & \text{otherwise} \end{cases}$$

9.3.2   M-step - Updating the Gating Network Parameters

When the Q-function is differentiated with respect to the gating network parameters, any contribution from the support parameters is considered a constant. Hence, the working Q-function for this maximisation, as a corollary of Theorem 6.6, is:

$$Q(\boldsymbol{\beta}, \overline{\boldsymbol{\beta}}) = -\frac{1}{2} \sum_{k=1}^{K} \sum_{i=1}^{M} \eta_{ik} \left[\chi_{ik} (\psi_{ik})^2 + \overline{\beta}_k \psi_{ik} \right]$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\chi}$ are the expected value of $\boldsymbol{z}$ and $\boldsymbol{\omega}$ respectively. This Q function is a quadratic form in $\boldsymbol{\psi}$, allowing a for a convenient maximisation routine:

$$\frac{\partial Q}{\partial \beta_k} = -\frac{1}{2} \sum_{i=1}^{M} \eta_{ik} \left[2 \, \chi_{ik} \, \psi_{ik} + \beta_k \right] x_i \; = \; 0$$

$$= \sum_{i=1}^{M} \eta_{ik} \left[\chi_{ik} (x_i^\top \beta_k - \log \left(\sum_{l \neq k} \exp(\underline{x}_i^T \underline{\beta}_k)) \right) - 0.5 \beta_k \right] x_i = 0$$

$$\implies$$

$$\sum_{i=1}^{M} \eta_{ik}\chi_{ik} \, x_i x_i^{\top} \, \beta_k = \sum_{i=1}^{M} \eta_{ik}\left(\chi_{ik}\log\left(\sum_{l\neq k}\exp(\underline{x}_i^{T}\underline{\beta}_k)\right) - 0.5\beta_k\right)x_i$$

Defining,

- $\Omega_k = \left(\eta_{1k}\chi_{1k}, \ldots, \eta_{Mk}\chi_{Mk}\right)$

- $\kappa_{ik} = \left(\eta_{1k}(\chi_{1k}\log\left(\sum_{l\neq k}\exp(\underline{x}_i^{T}\underline{\beta}_l)\right)) - 0.5\beta_{1k}), \ldots, \eta_{Mk}(\chi_{Mk}\log\left(\sum_{l\neq k}\exp(\underline{x}_i^{T}\underline{\beta}_l)\right)) - 0.5\beta_{Mk})\right)^{T}$

a closed form update is attained:

$$\hat{\beta}_k = \left(X^{\top}\Omega_k\,X\right)^{-1}X^{\top}\,\kappa_k.$$

If a Gaussian prior $\beta_k \sim \mathcal{N}(b, B)$ is used (MAP estimation), add B^{-1} to the left and $B^{-1}b$ to the right:

$$\hat{\beta}_k = \left(X^{\top}\Omega_k\,X + B^{-1}\right)^{-1}\left(X^{\top}\,\kappa_k + B^{-1}b\right).$$

9.4 Gibbs Sampling

In this model, full Bayesian inference is achieved using Gibbs Sampling, which is achievable given the closed form attained by having conjugate priors. The latent variables induced by Data Augmentation and the EM Algorithm, alongside the support parameters and the gating network parameters are sampled. Just like they were in the Mollica and Tardella model, the following parameters are sampled:

- **Sampling the Exponential Latent Variables y:**

  In the Partial Plackett-Luce density, the term $\sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q$ is replaced by the exponential random variable y_{it}, where:

$$y_{it} \sim \mathcal{E}\left(\sum_{j=1}^{N} p_j - \sum_{q=1}^{t-1} p_q\right)$$

  for judge i at preference level t. Thus, y_{it} is also sampled using this exponential augmentation during Gibbs Sampling.

- **Plackett-Luce support parameters p:**

  The support parameter for item j within expert k is sampled from a conditionally Gamma form:

$$p_{jk} \mid \cdot \sim \text{Gamma}\left(\sum_{i=1}^{M} z_{ik}\,u_{ij} + c_{jk}, d_k + A_{ik}\right)$$

- **Mixture Indicator Variables z:**

  The mixture indicator variables for the ith judge are distributed, conditional on the latest estimates for the mixture weights, by a multinomial case:

$$(z_{i1}, \ldots, z_{ik}, \ldots, z_{iK}) \sim \text{Multinomial}(1, \boldsymbol{\pi})$$
$$\sim \text{Multinomial}(1, \hat{\pi}_1, \ldots, \hat{\pi}_k, \ldots, \hat{\pi}_K)$$

  where

$$\hat{\pi}_k = \frac{\pi_k \mathbb{P}_{PPL}(R_i \mid \boldsymbol{p}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathbb{P}_{PPL}(R_i \mid \boldsymbol{p}_{k'})}$$

41

Also, the parameters involved in the inference for the gating networks regression weights have yet to be explained. They are:

- **Latent Pólya-Gamma random variables $\boldsymbol{\omega}$**:

When we introduce a Pólya-Gamma latent variable ω_{ik} to augment the multinomial-logit (softmax) gating term, we exploit the identity (Theorem6.4) that for any real ψ and $b > 0$,

$$\frac{(e^\psi)^a}{(1 + e^\psi)^b} = 2^{-b}\, e^{\kappa \psi} \int_0^\infty e^{-\frac{1}{2}\, \omega\, \psi^2}\, p(\omega)\, d\omega,$$

where $p(\boldsymbol{\omega})$ is the density of a Pólya-Gamma distributed random variable with parameters $(b, 0)$ and $\kappa = a - \frac{b}{2}$. Recall that our gating network is a *multinomial* logistic model over K experts:

$$\mathbb{P}(z_i = k \mid x_i) = \frac{\exp(\psi_{ik})}{\sum_{k'=1}^K \exp(\psi_{ik'})}, \quad \psi_{ik} = \beta_k^T x_i - c_i, \quad \text{with} \quad c_i = \log\left(\sum_{l \neq k} \exp(\beta_l^T x_i)\right)$$

It can be derived that conditional on the other $K - 1$ linear predictors $\{\psi_{ig}\}_{g \neq k}$, the probability of assigning observation i to expert k has the *binary* logistic form

$$\Pr(z_{ik} = 1 \mid x_i, \beta_{-k}) = \frac{\exp(\eta_{ik})}{1 + \exp(\psi_{ik})}, \quad \psi_{ik} = \psi_{ik} - \log \sum_{g \neq k} \exp(\psi_{ig}),$$

so that $z_{ik} \sim (p_{ik})$ with $p_{ik} = e^{\eta_{ik}}/(1 + e^{\eta_{ik}})$. By the Pólya-Gamma augmentation identity,

$$\frac{(e^\psi)^z}{(1 + e^\psi)^1} = 2^{-1} e^{(z - \frac{1}{2})\kappa} \int_0^\infty e^{-\frac{1}{2}\omega\, \eta^2}\, p(\omega)\, d\omega,$$

with $\omega \sim (1, 0)$ and $\boldsymbol{\kappa} = \boldsymbol{z} - \frac{1}{2}$. Hence the conditional distribution of $\boldsymbol{\omega}$ given $\boldsymbol{\psi}$ is

$$\omega_{ik} \mid \psi_{ik} \sim \left(1,\ \psi_{ik}\right).$$

- **Gating Network Mixing Parameters $\boldsymbol{\beta}$**:

In our fully-Bayesian MoE–PPL model, after data-augmenting with Pólya–Gamma variables $\{\omega_{ik}\}$, the conditional posterior for each expert's gating coefficient vector β_k is Gaussian. Specifically, recall that we have introduced

$$\psi_{ik} = \underline{\beta}_k^T \underline{x}_i - c_i, \quad c_i = \log \sum_{k' \neq k} \exp(\underline{\beta}_k^T \underline{x}_i),$$

and drawn

$$\omega_{ik} \sim (1, \psi_{ik}).$$

Writing X for the $M \times (L + 1)$ design matrix (including an intercept column) and defining

$$\Omega_k = \left(\eta_{1k}\chi_{1k}, \ldots, \eta_{Mk}\chi_{Mk}\right), \qquad \kappa_k = \left(\eta_{1k}(\tfrac{1}{2} + \chi_{1k}c_1), \ldots, \eta_{Mk}(\tfrac{1}{2} + \chi_{Mk}c_M)\right)^\top,$$

with $\eta_{ik} = \mathbb{E}[z_{ik}]$ and $\chi_{ik} = \mathbb{E}[\omega_{ik}]$. The Pólya-Gamma likelihood is conjugate to the normal distribution, hence the conditional posterior for β_k under a Gaussian prior $\beta_k \sim \mathcal{N}(b, B)$ is

$$\beta_k \mid \cdots \sim \left(V_k\, m_k,\ V_k\right),$$

where

$$V_k = \left(X^\top \Omega_k X + B^{-1}\right)^{-1}, \qquad m_k = X^\top \kappa_k + B^{-1}b.$$

The estimates are sampled iteratively based on the latest estimates for the other parameters.

The ideal amount of experts K is found using the Deviance Information Criterion (DIC). DIC uses Deviance, a goodness of fit statistic, which is penalised by the effective number of parameters p_D in a model. Formally, letting:

$$D(\theta) = -2\log L(\theta), \quad \bar{D} = \mathbb{E}[D(\theta) \mid R_i], \quad \hat{\theta}_{\mathrm{MAP}} = \arg\max_\theta \mathbb{P}(\theta \mid R_i)$$

with the effective number of parameters $p_D = \overline{D} - D(\hat{\theta}_{\mathrm{MAP}})$, then:

$$DIC_1 = \overline{D} + p_D$$

.

10 Case Study: 1997 Irish Presidential Election Opinion Poll

Gormley and Murphy's case study analyses opinion poll data collected early in the 1997 Irish presidential race [2]. Five candidates contested the election Banotti, McAleese, Nally, Roche, and Scallon - four women and one man. Irish Marketing Surveys interviewed 1 083 likely voters one month before polling day, and asked each respondent to rank as many of the five candidates as possible. Most voters supplied full rankings (or omitted at a single candidate) however, almost one-quarter revealed only partial rankings.

Alongside the respondents' rankings, the poll captured information on their age, urban / rural residence, gender, socioeconomic status, and satisfaction with the incumbent government . This allowed for the creation of seventeen covariates for each voter.

10.1 A Mixture of Experts Model for Ranked Data

In their work, Gormley and Murphy identified four latent voting blocs (i.e. expert networks) from the ranked and covariate data and using a Mixture of Experts model with a Benter ranking model. Their Benter dampening parameters were very close to 1 within their most optimal model, indicating that the likely voters made their ranking with high levels of certainty at each preference level. The
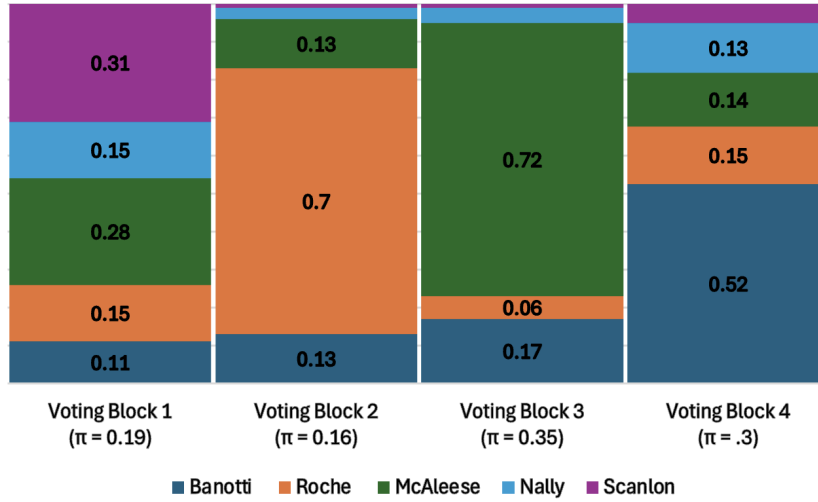

Figure 10.1: A Mosaic Plot of the 4 voting blocs. Component weights π are shown.

mixture of experts framework also gives a model based assessment of the factors that influence the membership of a voting bloc. In their best model, the age of a voter and their satisfaction with the current government are the best indicators:

- **Bloc 1 (conservative):** Higher support for Banotti and McAleese with older voters more likely members.

- **Bloc 2 (liberal):** Roche dominates and the log-odds indicate greater membership among younger voters.

- **Bloc 3 (pro-government):** Higher log-odds for voters satisfied with the incumbent government, consistent with McAleese's endorsement.

- **Bloc 4 (anti-government):** Membership odds are higher among voters dissatisfied with the government, which complies with the opposition candidate, Bannotti, being highly preferred.

44

10.2 A Mixture Model for Partially Ranked Data

Given the relatively high proportion of respondents that gave a partial ranking, it could be appropriate to attempt to fit the data using a partial Plackett-Luce model. The rankings were fitted to the mixture model of Mollica and Tardella [1].The DICM metric determined that having $K = 4$ mixture components best explained the data. Interestingly, although quite small, bloc 2 and 4 indicate very
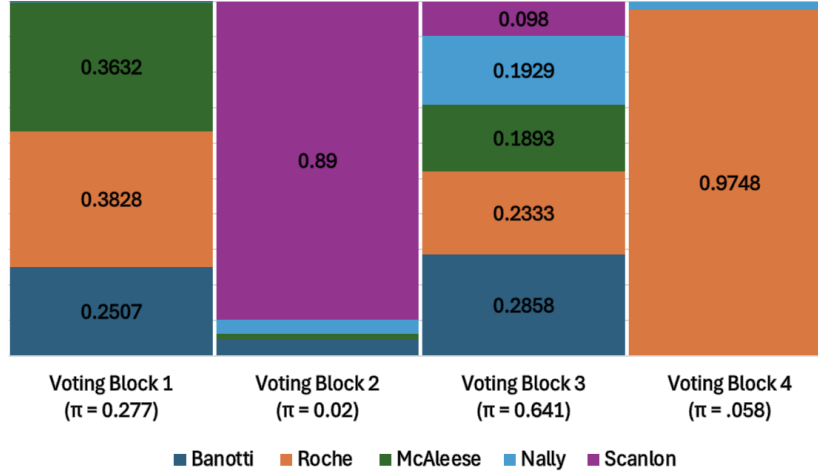

Figure 10.2: A Mosaic Plot of the 4 voting blocs. Component weights π are shown.

high support for only two candidates, Scanlon and Roche respectively. This could indicate that the supporters of those candidates, although they have comparatively low overall support, tend to only preference them and they are less likely to also rank other candidates. Such a phenomenon could only be uncovered by a model that incorporates partial ranking densities. That said, there is no model based way to infer the possible effects of a persons covariates on bloc membership.

10.3 A Bayesian Mixture of Experts Model for Partially Ranked Data

Now, the data is used in the Bayesian Mixture of Experts Model for Partially Ranked Data formulated in Section 9. First, the model was ran with all of the covariates used, and sequentially uninformative covariates were removed. Using this method, it was found that government satisfaction, age, and whether the voter lived in a city or not were the best predictors for bloc membership. The model uncovered five voting blocs.
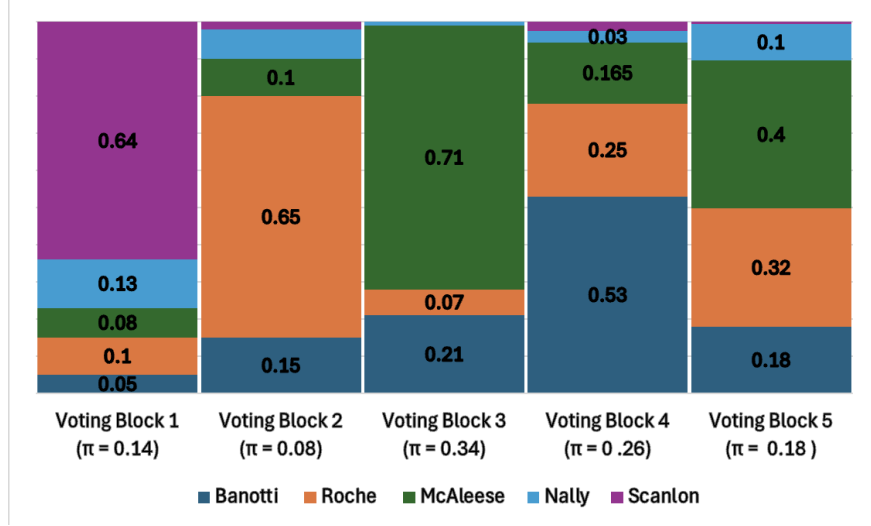
Figure 10.3: A Mosaic Plot of the five voting blocs. Component weights π are shown.

Owing to time constraints, no formal Gibbs sampling could be completed on these results, the overall support parameters align closely support for each candidate seen in the Gormley and Murphy paper, as well as the observed support parameter for the candidates among those polled. The gating network parameters suggest:

- **Voting bloc 1:** Perhaps there is also a conservative bloc in this model. Higher log-odds for both non-urban and older voters.

- **Voting bloc 2:** The log odds of being in this bloc are lower for younger voters.

- **Voting bloc 3:** There are large positive log-odds of being a member of this bloc for government supporters.

- **Voting bloc 4:** There appears to be an anti government cluster, given there are large drop off's in log-odds of being a member of this bloc for government supporters.

- **Voting bloc 5:** The log odds of being a member of this bloc are high for urban voters.

11 Conclusion

In the previous section, a well defined model was introduced that offers a promising approach to modelling partially ranked data. An attempt was made to apply this framework to exit poll data from the 2020 Irish General Election, but encountered a few challenges. First, the Irish National Election Study (INES), which usually receives public funding to collect voter data for each election, has struggled to garner sufficient backing, hence the quality of the data collected is deteriorating [38]. Although the framework presented in this report is theoretically suitable for ranked data, identifiability issues arise when the ratio of observed rankings is low. For the 2020 INES, at most four rankings were collected out of the eleven parties or groups of parties available for selection [39]. It is known that in a mixture of Plackett-Luce model with K components and full rankings of M items becomes unidentifiable when $K \geq \lceil (M+1)/2 \rceil$ [40]. Although this results pertains only to full rankings, it is plausible that in the case of identifiability, that models fitted to partial rankings are at least as hard identify than those fitted to complete rankings.

Second, up to 50% of the ballots included in the INES include 'ties' i.e. the voter gave a preference to more than one member of the same party in a particular constituency. In models for ranked data, ties are usually removed, and/or replaced randomly with an unranked item. In the case of elections with a ranked choice voting systems, ties can be generally allowed if more than one of the candidates are members of the same party. The removal of such rankings can remove data that is otherwise highly explanatory of the behaviour of voters. Perhaps a hierarchical structure or a mixture of experts model with item based covariates could be explored to reduce the need for the removal of preferences. [41] [42]. Alternatively, nonparametric approaches like the Borda count model could be used [43].

Finally, it could be fruitful to develop external model-comparison metrics—beyond internal criteria such as BIC or DICM to evaluate different modeling approaches. Additionally, identifying the most informative covariates to begin with or as a part of the model, rather than iteratively including and excluding predictors based on perceived influence, could improve both efficiency and validity.

References

[1]   Cristina Mollica and Luca Tardella. "Bayesian Plackett–Luce mixture models for partially ranked data". In: *Psychometrika* 82.2 (2017), pp. 442–458.

[2]   Isobel Claire Gormley and Thomas Brendan Murphy. "A mixture of experts model for rank data with applications in election studies". In: *The Annals of Applied Statistics* 2.4 (Dec. 2008). ISSN: 1932-6157. DOI: 10.1214/08-aoas178. URL: http://dx.doi.org/10.1214/08-AOAS178.

[3]   Benjamin Craig, J.J.V. Busschbach, and Joshua Salomon. "Modeling Ranking, Time Trade-Off, and Visual Analog Scale Values for EQ-5D Health States: A Review and Comparison of Methods". In: *Medical care* 47 (June 2009), pp. 634–41. DOI: 10.1097/MLR.0b013e31819432ba.

[4]   Regina Dittrich, W. Katzenbeisser, and Heribert Reisinger. "The analysis of rank ordered preference data based on Bradley-Terry Type ModelsDie Analyse von Präferenzdaten mit Hilfe von log-linearen Bradley-Terry Modellen". In: *Or Spektrum* 22 (Feb. 2000), pp. 117–134. DOI: 10.1007/s002910050008.

[5]   William Benter. "Computer Based Horse Race Handicapping and Wagering Systems: A Report". In: *Efficiency of Racetrack Betting Markets*, pp. 183–198. DOI: 10.1142/9789812819192_0019. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812819192_0019. URL: https://www.worldscientific.com/doi/abs/10.1142/9789812819192_0019.

[6]   J. I. Marden. *Analyzing and Modeling Rank Data*. 1st. Chapman and Hall/CRC, 1995. DOI: 10.1201/b16552. URL: https://doi.org/10.1201/b16552.

[7]   L. L. Thurstone. "A Law of Comparative Judgment". In: *Psychological Review* 34.4 (1927), pp. 273–286. DOI: 10.1037/h0070288. URL: https://doi.org/10.1037/h0070288.

[8]   B. Babington Smith. "Discussion of Professor Ross's Paper". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 12 (1950), pp. 53–56.

[9]   Ralph Allan Bradley and Milton E. Terry. "Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2334029 (visited on 04/13/2025).

[10] R. L. Plackett. "The Analysis of Permutations". In: *Journal of the Royal Statistical Society* (1975).

[11] John Guiver and Edward Snelson. "Bayesian inference for Plackett-Luce ranking models". In: vol. 382. June 2009, p. 48. DOI: 10.1145/1553374.1553423.

[12] Robert Jacobs et al. "Adaptive Mixture of Local Expert". In: *Neural Computation* 3 (Feb. 1991), pp. 78–88. DOI: 10.1162/neco.1991.3.1.79.

[13] Weilin Cai et al. *A Survey on Mixture of Experts*. 2024. arXiv: 2407.06204 [cs.LG]. URL: https://arxiv.org/abs/2407.06204.

[14] Isobel Claire Gormley and Thomas Brendan Murphy. "Mixture of experts modelling with social science applications". In: *Mixtures: Estimation and applications* (2011), pp. 101–121.

[15] R.J. Larsen and M.L. Marx. *An Introduction to Mathematical Statistics and Its Applications*. v. 1. Pearson Prentice Hall, 2006. ISBN: 9780131867932. URL: https://books.google.ie/books?id=4CIYHAAACAAJ.

[16] In Jae Myung. "Tutorial on maximum likelihood estimation". In: *Journal of Mathematical Psychology* 47.1 (2003), pp. 90–100. ISSN: 0022-2496. DOI: https://doi.org/10.1016/S0022-2496(02)00028-7. URL: https://www.sciencedirect.com/science/article/pii/S0022249602000287.

[17] David R. Hunter and Kenneth Lange. "Quantile Regression via an MM Algorithm". In: *Journal of Computational and Graphical Statistics* 9.1 (2000), pp. 60–77. ISSN: 10618600. URL: http://www.jstor.org/stable/1390613.

[18] Kenneth Lange and Hua Zhou. "A Legacy of EM Algorithms". In: *International Statistical Review* 90.S1 (2022), S52–S66. DOI: https://doi.org/10.1111/insr.12526. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/insr.12526. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12526.

[19] David R Hunter and Kenneth Lange and. "A Tutorial on MM Algorithms". In: *The American Statistician* 58.1 (2004), pp. 30–37. DOI: 10.1198/0003130042836. eprint: https://doi.org/10.1198/0003130042836. URL: https://doi.org/10.1198/0003130042836.

[20] Dankmar Boehning and Bruce Lindsay. "Monotonicity of quadratic-approximation algorithms". In: *Annals of the Institute of Statistical Mathematics* 40 (Feb. 1988), pp. 641–663. DOI: 10.1007/BF00049423.

[21] A. Albert and J. A. Anderson. "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models". In: *Biometrika* 71.1 (1984), pp. 1–10. ISSN: 00063444. URL: http://www.jstor.org/stable/2336390 (visited on 03/31/2025).

[22] Duc Nguyen and Anderson Zhang. *Efficient and Accurate Learning of Mixtures of Plackett-Luce Models.* Feb. 2023. DOI: 10.48550/arXiv.2302.05343.

[23] Lucas Maystre and Matthias Grossglauser. "Fast and accurate inference of Plackett-Luce models". In: *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1.* NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 172–180.

[24] Suorong Yang et al. *Image Data Augmentation for Deep Learning: A Survey.* 2023. arXiv: 2204.08610 [cs.CV]. URL: https://arxiv.org/abs/2204.08610.

[25] Jim Albert and Siddhartha Chib. "Bayesian Analysis of Binary and Polychotomous Response Data". In: *Journal of The American Statistical Association - J AMER STATIST ASSN* 88 (June 1993), pp. 669–679. DOI: 10.1080/01621459.1993.10476321.

[26] Nicholas G. Polson, James G. Scott, and Jesse Windle. *Bayesian inference for logistic models using Polya-Gamma latent variables.* 2013. arXiv: 1205.0310 [stat.ME]. URL: https://arxiv.org/abs/1205.0310.

[27] F. W. J. Olver et al., eds. *NIST Digital Library of Mathematical Functions.* Version Release 1.2.4. NIST Digital Library of Mathematical Functions. Mar. 15, 2025. URL: https://dlmf.nist.gov/.

[28] James G. Scott and Liang Sun. *Expectation-maximization for logistic regression.* 2013. arXiv: 1306.0040 [stat.CO]. URL: https://arxiv.org/abs/1306.0040.

[29] MIT OpenCourseWare. *Lecture 14: Poisson Process – I: Competing Exponentials.* https://ocw.mit.edu. Lecture video for 6.041SC Probabilistic Systems Analysis and Applied Probability (Fall 2013). Instructor: Prof. John Tsitsiklis; presented by Jimmy Li. Retrieved from https://ocw.mit.edu. 2013.

[30] Francois Caron and Arnaud Doucet. *Efficient Bayesian Inference for Generalized Bradley-Terry Models.* 2010. arXiv: 1011.1761 [stat.ME]. URL: https://arxiv.org/abs/1011.1761.

[31] Panagiotis Papastamoulis. "label.switching: An R Package for Dealing with the Label Switching Problem in MCMC Outputs". In: *Journal of Statistical Software, Code Snippets* 69.1 (2016), pp. 1–24. DOI: 10.18637/jss.v069.c01. URL: https://www.jstatsoft.org/index.php/jss/article/view/v069c01.

[32] I. C. Gormley and T. B. Murphy. "Exploring voting blocs within the Irish electorate: A mixture modeling approach". In: *Journal of the American Statistical Association* 103 (2008), pp. 1014–1027.

[33] David R. Hunter. "MM algorithms for generalized Bradley-Terry models". In: *The Annals of Statistics* 32.1 (2004), pp. 384–406. DOI: 10.1214/aos/1079120141. URL: https://doi.org/10.1214/aos/1079120141.

[34] Isobel Gormley and Thomas Murphy. "Clustering ranked preference data using sociodemographic covariates". In: *Choice Modelling: The State-of-the-Art and the State-of-Practice* (Jan. 2010).

[35] Isobel Gormley and Thomas Murphy. "Exploring Voting Blocs Within the Irish Electorate: A Mixture Modeling Approach". In: *Journal of the American Statistical Association* 103 (Sept. 2008), pp. 1014–1027. DOI: 10.1198/016214507000001049.

[36] Xiao-Li Meng and Donald Rubin. "Maximum Likelihood Estimation via the ECM Algorithm: A General Framework". In: *Biometrika* 80 (June 1993). DOI: 10.1093/biomet/80.2.267.

[37] Gideon Schwarz. "Estimating the Dimension of a Model". In: *Annals of Statistics* 6.2 (July 1978), pp. 461–464.

[38] Johan A. Elkink and David M. Farrell and. "Predicting vote choice in the 2020 Irish general election". In: *Irish Political Studies* 36.4 (2021), pp. 521–534. DOI: 10.1080/07907184.2021.1978219. eprint: https://doi.org/10.1080/07907184.2021.1978219. URL: https://doi.org/10.1080/07907184.2021.1978219.

[39] Johan Elkink and David Farrell. *2020 UCD Online Election Poll (INES 1)*. Version V4. 2020. DOI: 10.7910/DVN/E6TAVY. URL: https://doi.org/10.7910/DVN/E6TAVY.

[40] Zhibing Zhao, Peter Piech, and Lirong Xia. "Learning Mixtures of Plackett-Luce models". In: *CoRR* abs/1603.07323 (2016). arXiv: 1603.07323. URL: http://arxiv.org/abs/1603.07323.

[41] Maksim Tkachenko and Hady W. Lauw. "Plackett-Luce Regression Mixture Model for Heterogeneous Rankings". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016, pp. 237–246. ISBN: 9781450340731. DOI: 10.1145/2983323.2983763. URL: https://doi.org/10.1145/2983323.2983763.

[42] Paul D. Allison and Nicholas A. Christakis. "Logit Models for Sets of Ranked Items". In: *Sociological Methodology* 24 (1994), pp. 199–228. ISSN: 00811750, 14679531. URL: http://www.jstor.org/stable/270983 (visited on 04/19/2025).

[43] Duc Nguyen. *Efficient and Accurate Top-K Recovery from Choice Data*. 2022. arXiv: 2206.11995 [cs.LG]. URL: https://arxiv.org/abs/2206.11995.

[44] Cristina Mollica and Luca Tardella. "PLMIX Version 2 for R: Bayesian analysis of finite mixtures of Plackett-Luce models for partially ranked data". In: *Psychometrika* 82.2 (2017), pp. 442–458. ISSN: 0033-3123. DOI: 10.1007/s11336-016-9530-0.

A A Mixture Of Experts Model For Rank Data Derivations

A.1 Expression for π_{ik}

Recall that:

$$\log\left(\frac{\pi_{ik}}{\pi_{i1}}\right) = \beta_{k0} + \beta_{k1}\omega_{i1} + \beta_{k2}\omega_{i2} + \cdots + \beta_{kL}\omega_{iL}$$

$$= \underline{\beta}_k^T \underline{\omega}_i$$

Exponentiating both sides:

$$\frac{\pi_{ik}}{\pi_{i1}} = \exp(\underline{\beta}_k^T \underline{\omega}_i)$$

$$\pi_{ik} = \pi_{i1} \exp(\underline{\beta}_k^T \underline{\omega}_i)$$

$$\sum_{k=1}^{K} \pi_{ik} = \pi_{i1} \sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{\omega}_i)$$

$$1 = \pi_{i1} \sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{\omega}_i)$$

$$\pi_{i1} = \frac{1}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{\omega}_i)}$$

subbing back in:

$$\pi_{ik} = \frac{\exp(\underline{\beta}_k^T \underline{\omega}_i)}{\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{\omega}_i)}$$

A.2 Expression for z_{ik}

Recalling Bayes Theorem:

$$\mathbb{P}(B_K|A) = \frac{\mathbb{P}(A|B_K)\mathbb{P}(B_K)}{\sum_i \mathbb{P}(A|B_i)\mathbb{P}(B_i)}$$

Hence, if B_K represents the event of of observation i belonging to expert J (given the parameters p & α), and A represents the $\underline{x}_i$'th observation, then the estimate for z_{ik} can be derived as:

$$\hat{z}_{ik} = \mathbb{P}(B_K|\underline{x}_i)$$

$$= \frac{\mathbb{P}(\underline{x}_i|B_K)\mathbb{P}(B_K)}{\sum_{k'=1}^{K} \mathbb{P}(\underline{x}_i|B_{k'})\mathbb{P}(B_{k'})}$$

$$= \frac{\mathbb{P}(\underline{x}_i|\underline{p}_k, \alpha_t)\pi_{ik}}{\sum_{k'=1}^{K} \mathbb{P}(\underline{x}_i|\underline{p}_{k'}, \alpha_t)\pi_{ik'}}$$

A.3 Maximization with respect to the Benter support parameters

Under the Expectation Conditional Maximization framework, any terms in the $Q-$function will become constants, and will be cancelled when differentiated. Hence, the modified $Q-$function for the

step that is maximising with respect to the Benter support parameters is:

$$q = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t} \right]$$

where $\overline{\alpha}_t$ is the latest estimation for α_t from the previous iteration of the EMM algorithm.

Here, the term $-\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}$ is problematic as it may not always be differentiable, so a minorizing function is constructed. Given that the EMM algorithm ensures monotonically increasing estimates for the parameters, an appropriate lower bound is constructed around the most recent estimate of $\underline{p}_k$, which is defined as $\overline{p}_{kj}$. Recall that, for a convex function $f(y)$:

$$f(y) \geq f(x) + f'(x)(y - x)$$

Where x is an appropriately chosen value to be a lower bound. In the case of the EMM algorithm, the estimation of the target parameter from the previous iteration is typically chosen.

The function to be optimised in this case is $-\log(y)$, which is convex. Therefore:

$$-\log(y) \geq -\log(x) + [\log(x)]'(y - x)$$
$$\geq -\log(x) + \frac{1}{y}(y - x)$$
$$\geq -\log(x) - \frac{y}{x}$$

$$\implies$$

$$-\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t} \geq \log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} + 1 - \frac{\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}}{\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}}$$

with $\overline{p}_{kj}$ representing the estimate for the Benter support parameter from the previous iteration. In this inequality, the right hand side is a minorizing surrogate function of $-\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}$. It follows that:

$$q \geq \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \left(\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} + 1 - \frac{\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}}{\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$
$$\geq \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \sum_{t=1}^{n_i} \left(\frac{\log \sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}_t}}{\log \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$

A further surrogate function is constructed to deal with the term $-\sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}}$. Utilising the monotonically increasing estimates condition of the EMM algorithm, $(\overline{p}_{kj})$ can be utilised to show that:

$$-\sum_{s=t}^{N} p_{kc(i,s)}^{\overline{\alpha}} \geq -\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} - \sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} (p_{kc(i,s)} - \overline{p}_{kc(i,s)})$$
$$\geq -\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} + \overline{\alpha}_t \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} - \overline{\alpha}_t \sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t -} p_{kc(i,s)}$$

The final surrogate function can be constructed as:

$$Q \geq q = \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \left(\frac{\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} p_{kc(i,s)}}{\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right) \right]$$

$$\geq q = \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \overline{\alpha}_t \log p_{kc(i,t)} - \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} p_{kc(i,s)} \right) \right]$$

up to a constant.

Now, the final $q-$function is differentiable. It is differentiated with respect to p_{kj}:

$$\frac{\partial q}{\partial p_{kj}} = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\frac{\overline{\alpha}_t}{p_{kc(i,t)}} \mathbb{1}_{j=c(i,t)} - \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right) \right]$$

In this derivative $\mathbb{1}_{j=c(i,s)}$ is an indicator function. The Benter support parameter p_{kj} is only influenced by the voters in expert network k that actually give a preference to candidate j. Hence, the function:

$$\mathbb{1}_{j=c(i,s)} = \begin{cases} 1 & \text{voter } i \text{ gives candidate } j \text{ their } s \text{ preference level} \\ 0 & \text{otherwise} \end{cases}$$

is defined.

Now, $\dfrac{\partial q}{\partial p_{kj}}$ is set to 0, and a closed form for $\hat{p}_{kj}$ is found:

$$0 = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\frac{\overline{\alpha}_t}{p_{kj}} \mathbb{1}_{j=c(i,t)} - \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right) \right]$$

$$= \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\frac{\overline{\alpha}_t}{p_{kj}} \mathbb{1}_{j=c(i,t)} \right] - \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right) \right]$$

$$\implies \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \frac{\overline{\alpha}_t}{p_{kj}} \mathbb{1}_{j=c(i,t)} = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right)$$

For ease of notation, ω_{kj} is defined:

$$\omega_{kj} = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \overline{\alpha}_t \mathbb{1}_{j=c(i,t)}$$

Therefore:

$$\frac{\omega_{kj}}{p_{kj}} = \sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right)$$

$$\hat{p}_{kj} = \frac{\omega_{kj}}{\sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \mathbb{1}_{j=c(i,s)} \right)}$$

There is one case where this maximised equation is not accurate. if a voter ranks $n_i = N - 1$ out of N candidates, they are still in practice giving the only candidate they did not give a ranking to a preference. A new indicator function δ is defined:

$$
\delta_{ijs} = \begin{cases} 1 & \text{if } j = r_{(i,s)} \text{ for } 1 \leq s \leq n_i \\ 1 & \text{if } j \neq r_{(i,l)} \text{ for } 1 \leq l \leq n_i, \text{ but } s = N + 1 \\ 0 & \text{otherwise} \end{cases}
$$

which appropriately addresses these specific cases. Hence, the closed form for the maximised estimate of the Benter support parameter of candidate j in expert network k is:

$$
\hat{p}_{kj} = \frac{\omega_{kj}}{\sum_{i=1}^{M} \sum_{t=1}^{n_i} \hat{z}_{ik} \left(\sum_{s=t}^{N} \overline{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum_{s=t}^{N+1} \overline{\alpha}_t \overline{p}_{kc(i,s)}^{\overline{\alpha}_t - 1} \delta_{ijs} \right)}
$$

A.4   Maximization with respect to the Benter dampening parameters

Conditional on the estimate of $\hat{p}_{kj}$ gained in the previous step, an appropriate surrogate function is defined and then a maximised estimate of α_t is derived. Recall that the terms of interest in the $Q-$function are:

$$
q = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \alpha_t \log \hat{p}_{kc(i,t)} - \log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha_t} \right]
$$

with the previous iteration's estimate for p remaining as a constant. The term $- \log \sum_{s=t}^{M} \hat{p}_{kc(i,s)}$ is not differential due to its combinatorial nature. It is a convex function, so again the inequality:

$$
- \log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha_t} \geq - \log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} + 1 - \frac{\log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha_t}}{\log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t}}
$$

is used to form a minorizing function:

$$
\begin{aligned}
Q \geq q &= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \alpha_t \log \hat{p}_{kc(i,t)} - \left(\log \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha} + 1 - \frac{\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha}}{\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}}} \right) \right] \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\sum_{t=1}^{n_i} \alpha_t \log \hat{p}_{kc(i,t)} + \sum_{t=1}^{n_i} \left(\frac{- \sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\alpha}}{\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}}} \right) \right]
\end{aligned}
$$

The term $- \hat{p}_{kc(i,s)}^{\alpha}$ is not differentiable. Given that α_t takes on values between zero and one, $- \hat{p}^{\alpha_t}$ is a concave function. Hence, if the function $g(\alpha_t) = - \hat{p}^{\alpha_t}$ is defined, it can bounded around an appropriately chosen y:

$$
g(y) \leq g(x) + g'(x)^t (y - x) + \frac{1}{2} (y - x)^t \boldsymbol{B} (y - x)
$$

An appropriate $\boldsymbol{B}$ must be chosen, such that $\boldsymbol{H} < \boldsymbol{B}$, where $\boldsymbol{H}$ is the Hessian Matrix. In this case, $g''(\alpha_t) = - \hat{p}^{\alpha_t} [\log(\hat{p})]^2$, hence a suitable upper bound for the Hessian Matrix is $\log(\hat{p})^2$. Therefore:

$$
- \hat{p}^{\alpha_t} \geq - \hat{p}^{\overline{\alpha}_t} - (\log \hat{p}) \hat{p}^{\overline{\alpha}_t} (\alpha_t - \overline{\alpha}_t) - \frac{1}{2} (\alpha_t - \overline{\alpha}_t)^2 (\log \hat{p})^2
$$

$$\Longrightarrow$$

$$
\begin{aligned}
Q \geq q &= \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\alpha_t \log \hat{p}_{kc(i,t)} + \frac{- \sum_{s=t}^{N} \left(\hat{p}^{\overline{\alpha}_t} - (\log \hat{p}) \hat{p}^{\overline{\alpha}_t} (\alpha_t - \overline{\alpha}_t) - \frac{1}{2} (\alpha_t - \overline{\alpha}_t)^2 (\log \hat{p})^2 \right)}{\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t}} \right] \\
&= \sum_{i=1}^{M} \sum_{k=1}^{K} \sum_{t=1}^{n_i} \hat{z}_{ik} \left[\alpha_t \log \hat{p}_{kc(i,t)} + \left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}} \right)^{-1} \left(\sum_{s=t}^{N} \left\{ - (\log \hat{p}) \hat{p}^{\overline{\alpha}} (\alpha - \overline{\alpha}) - \frac{1}{2} (\alpha - \overline{\alpha})^2 (\log \hat{p})^2 \right\} \right) \right]
\end{aligned}
$$

54

Now, the final $q-$function is differentiable. It is differentiated with respect to α_t:

$$\frac{\partial q}{\partial \alpha_t} = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log \hat{p}_{kc(i,t)} + \left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \right.$$
$$\left. \cdot \left(\sum_{s=t}^{N} \left\{ -(\log \hat{p}_{kc(i,s)}) \hat{p}_{kc(i,s)}^{\overline{\alpha}} - (\alpha_t - \overline{\alpha}_t)(\log \hat{p}_{kc(i,s)})^2 \right\} \right) \right] \cdot \mathbb{1}_{t \leq n_i}$$

$$0 = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log \hat{p}_{kc(i,t)} \right]$$
$$+ \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \cdot \left\{ \sum_{s=t}^{N} \left(-(\log \hat{p}_{kc(i,s)}) \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right) - \alpha_t \sum_{s=t}^{N} (\log \hat{p}_{kc(i,s)})^2 + \overline{\alpha}_t (\log \hat{p}_{kc(i,s)})^2 \right\} \right] \cdot \mathbb{1}_{t \leq n_i}$$

$$= \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\log \hat{p}_{kc(i,t)} \right] + \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left\{ \left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \cdot \left(\sum_{s=t}^{N} -(\log \hat{p}_{kc(i,s)}) \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right) \right\}$$
$$- \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left\{ \left(\sum_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} - \alpha_t \sum_{s=t}^{N} (\log \hat{p}_{kc(i,s)})^2 + \overline{\alpha}_t (\log \hat{p}_{kc(i,s)})^2 \right\}$$

$$\implies$$

$$\hat{\alpha}_t = \frac{\sum\limits_{i=1}^{M} \sum\limits_{k=1}^{K} \hat{z}_{ik} \left[\log \hat{p}_{kc(i,s)} + \left(\sum\limits_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum\limits_{s=t}^{N} -(\log \hat{p}_{kc(i,s)}) \hat{p}_{kc(i,s)}^{\overline{\alpha}} + \overline{\alpha}_t (\log \hat{p}_{kc(i,s)})^2 \right) \right] \cdot \mathbb{1}_{t \leq n_i}}{\sum\limits_{i=1}^{M} \sum\limits_{k=1}^{K} \hat{z}_{ik} \left[\left(\sum\limits_{s=t}^{N} \hat{p}_{kc(i,s)}^{\overline{\alpha}_t} \right)^{-1} \left(\sum\limits_{s=t}^{N} (\log \hat{p}_{kc(i,s)})^2 \right) \right] \cdot \mathbb{1}_{t \leq n_i}}$$

where the indicator function $\mathbb{1}_{t \leq n_i}$ is defined such that:

$$\mathbb{1}_{t \leq n_i} = \begin{cases} 1 & \text{voter } i \text{ expresses a preference for candidate } t \\ 0 & \text{otherwise} \end{cases}$$

A.5   Maximization with respect to the gating network parameters

In this section, the gating network parameters will be maximised. The gating network values are determined using multinomial logistic function, where the independent variables are the covariates of voter i. Hence, the log likelihood of the gating paremeter for voter i's membership of expert network k is:

$$\log \left(\frac{\pi_{ik}}{\pi_{i1}} \right) = \beta_{k0} + \beta_{k1} \omega_{i1} + \beta_{k2} \omega_{i2} + \cdots + \beta_{kL} \omega_{iL}$$
$$= \underline{\beta}_k^T \underline{x}_i$$

where $\underline{\beta}_k^T$ are the gating network parameters. For this step, the target $Q-$function with respect to β is:

$$q = \sum_{i=1}^{M} \sum_{k=1}^{K} \hat{z}_{ik} \left[\exp(\underline{\beta}_k^T \underline{x}_i) - \log \left(\sum_{k=1}^{K} \exp(\underline{\beta}_k^T \underline{x}_i) \right) \right]$$

up to a constant. Direct maximization of this multinomial-logit can be difficult, hence a minorization-maximisation approach is employed. Specifically, a local quadratic surrogate function around the

previous iterations estimate of the gating parameters $(\underline{\beta}_k^{(h)})$ is constructed:

$$q(\underline{\beta}_k^{(h)}) + q'(\underline{\beta}_k^{(h)})^T(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) + \frac{1}{2}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})^T \boldsymbol{B}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})$$

A suitable matrix $\boldsymbol{B}$ should be chosen, where $\boldsymbol{B}$ bounds the Hessian Matrix $\boldsymbol{H}(\underline{\beta}_k^{(h)})$ from below. The highest possible value for $\underline{\beta}_k^{(h)}$ is attained when there are two independent covariates ω_1 & ω_2, with one independent outcome variable Y_i:

$$Y_i = \beta_1 \omega_{i1} + \beta_2 \omega_{i2}$$

with likelihood function:

$$L(\beta_1, \beta_2) = \prod_{i=1}^{n} \left[\pi_i(\underline{\beta})^{Y_i} \left[1 - \pi_i(\underline{\beta}) \right] \right]^{1-Y_i}$$

and log-likelihood function:

$$\ell(\beta_1, \beta_2) = \sum_{i=1}^{n} \left((Y_i) \log[\pi_i(\underline{\beta})] + (1 - Y_i) \log[1 - \pi_i(\underline{\beta})] \right)$$

The second partial derivative, with respect to the gating network parameters is:

$$\nabla^2 \ell(\beta_1, \beta_2) = -\sum_{i=1}^{n} \pi_i(\beta) \left[1 - \pi_i(\beta) \right] \begin{pmatrix} \omega_{i1} \\ \omega_{i2} \end{pmatrix} \begin{pmatrix} \omega_{i1} & \omega_{i2} \end{pmatrix}.$$

$\pi_i(\beta) \left[1 - \pi_i(\beta) \right]$ is maximised when $\pi_i(\beta) = \frac{1}{4}$, hence the negative definite matrix $\boldsymbol{B} = -\frac{1}{4} \sum_{i=1}^{M} \underline{x}_i \underline{x}_i^T$ is defined to derive the minorizing surrogate function:

$$q(\underline{\beta}_k^{(h)}) + q'(\underline{\beta}_k^{(h)})^T(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) + \frac{1}{2}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})^T \left(-\frac{1}{4} \sum_{i=1}^{M} \underline{x}_i \underline{x}_i^T (\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) \right)$$

around the point $\underline{\beta}_k^{(h)}$. Now, this surrogate function is maximised:

$$\frac{\partial}{\partial \beta} \left[q(\underline{\beta}_k^{(h)}) + q'(\underline{\beta}_k^{(h)})^T(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) + \frac{1}{2}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})^T \left(\boldsymbol{B}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)}) \right) \right] = q'(\underline{\beta}_k^{(h)}) + \boldsymbol{B}(\underline{\beta}_k^{(h+1)} - \underline{\beta}_k^{(h)})$$

$$\implies$$

$$\underline{\beta}_k^{(h+1)} = \underline{\beta}_k^{(h)} - \boldsymbol{B}^{-1} q'(\underline{\beta}_k^{(h)})$$

is the formula that defines the value of the next iteration, in terms of the estimate for the gating network parameters derived at the previous step.

B   Code

The code used to implement the new model introduced in this paper was heavily influenced by the code of Mollica and Tardella model presented in Section 7 [1]. The authors of the paper created an R package with various C++ exports to allow for the replication of their work [44]. The main piece of code, the Maximum a Posteriori (MAP) estimation of the novel Bayesian Mixture of Experts Model is presented here:

```
1 library(Rcpp)
2 library(RcppArmadillo)
3 source("~/FYP_R_Code/as.top_ordering.R")
4 source("~/FYP_R_Code/fill_single_entries.R")
5 source("~/FYP_R_Code/bicPLMIX.R")
6 sourceCpp("/Users/benheskin/FYP_R_Code/FYP_R_Code/howmanyranked.cpp")
```

```
7  sourceCpp("~/FYP_R_Code/UpPhetV0.cpp")
8  sourceCpp("/Users/benheskin/FYP_R_Code/FYP_R_Code/umat.cpp")
9  sourceCpp("/Users/benheskin/FYP_R_Code/FYP_R_Code/UpWhet.cpp")
10 sourceCpp("/Users/benheskin/FYP_R_Code/FYP_R_Code/Estep_z.cpp")
11 sourceCpp("~/FYP_R_Code/Estep_omega.cpp")
12 sourceCpp("~/FYP_R_Code/FYP_R_Code/UpPhetpartial.cpp")
13 sourceCpp("~/FYP_R_Code/upPis.cpp")
14 sourceCpp("~/FYP_R_Code/CompRateYpartial.cpp")
15 sourceCpp("~/FYP_R_Code/estepV0.cpp")
16 sourceCpp("~/FYP_R_Code/BetaV0.cpp")
17 sourceCpp("~/FYP_R_Code/CompRateP.cpp")
18 sourceCpp("~/FYP_R_Code/pihatV0.cpp")
19 sourceCpp("~/FYP_R_Code/upPsi.cpp")
20 sourceCpp("~/FYP_R_Code/UpPgOmega.cpp")
21 sourceCpp("~/FYP_R_Code/UpKappas.cpp")
22 sourceCpp("~/FYP_R_Code/UpC.cpp")
23 sourceCpp("~/FYP_R_Code/UpDot.cpp")
24 sourceCpp("~/FYP_R_Code/SPV0.cpp")
25 sourceCpp("~/FYP_R_Code/loglikPLMIXoE.cpp")
26
27
28 if(class(pi_inv)[1]!="top_ordering"){
29   if(class(pi_inv)[1]=="RankData"){
30     pi_inv=as.top_ordering(data=pi_inv)
31   }
32   if(class(pi_inv)[1]=="rankings"){
33     pi_inv=as.top_ordering(data=pi_inv)
34   }
35   if(class(pi_inv)[1]=="matrix" | class(pi_inv)[1]=="data.frame"){
36     pi_inv=as.top_ordering(data=pi_inv,format_input="ordering",aggr=FALSE)
37   }
38 }
39
40
41
42 pi_inv <- fill_single_entries(data=pi_inv)
43 N <- nrow(pi_inv)
44 M <- N
45 n_rank <-  howmanyranked(pi_inv)
46 K <- ncol(pi_inv)
47 L <- ncol(X)
48 mapMoPLMIX <- function(pi_inv,G,K,
49                        n_iter=500,
50                        hyper=NULL,
51                        eps=10^(-6),
52                        centered_start=FALSE,
53                        plot_objective=FALSE, X ){
54
55
56
57   u_bin <- umat(pi_inv)
58   rho   <- matrix(1:K, nrow=G, ncol=K, byrow=TRUE)
59   n_rank<- howmanyranked(pi_inv)
60
61   L <- ncol(X)
62
63   if (!is.null(hyper)) {
64     shape0 <- hyper$shape0
65     rate0  <- hyper$rate0
66   } else {
67     shape0 <- matrix(1, nrow=G, ncol=K)
68     rate0  <- rep(0, G)
69   }
70
71
72
73
74
```

57

```r
  # creates a reference ordering that has 1's in column 1, 2's in column 2.


  ref_known <- TRUE
  ref_vary <- FALSE

  PriorCovariance <- diag(.01, L, L)

  upBhet <- function(X, Omega, kappa, PriorCovariance){
    B_inv <- PriorCovariance
    S <- t(X) %*% Omega %*% X +B_inv
    d <- t(X) %*% kappa
    beta_k <- solve(S,d)
    return(beta_k)
  }




  betaT <- as.matrix(BetaV0(G, ncol(X)))
  piks  <- upPis(X=X, betaT=betaT)

  ParV0 <- function(
    pi_inv  = pi_inv,
    G       = G,
    K       = K,
    n_rank  = n_rank,
    u_bin   = u_bin,
    shape0  = shape0,
    rate0   = rate0,
    rho     = rho,
    iterations = 500
  ) {

    p <- matrix(1/K, nrow=G, ncol=K)
    p <- p / rowSums(p)
    z <- matrix(runif(N * G), nrow = N, ncol = G)
    z <- z / rowSums(z)
    piV0 <- runif(G)
    piV0 <- piV0 / sum(piV0)
    piks  <- upPis(X = X, betaT = betaT)
    betaT <- as.matrix(BetaV0(G, L))
    # store initialised stuff



    for(l in seq_len(iterations)){

      ll <- estepV0(p = p, pi_inv = pi_inv, piV0 = piV0, n_rank = n_rank, z = z)
      p <- UpPhetV0(p         = p,
                    ref_order = rho,
                    pi_inv    = pi_inv,
                    u_bin     = u_bin,
                    z_hat     = z,
                    shape0    = shape0,
                    rate0     = rate0,
                    n_rank    = n_rank)

      p <- p / rowSums(p)

      pi_mat <- matrix(rep(piV0, each = N), nrow = N, ncol = G)
      pi_mat <- pihatV0(pi_mat, z)
      piV0 <- pi_mat[1, ]

    }

    list(
      p     = p,
```

```r
143        piV0 = piV0,
144        z    = z
145      )
146
147    }
148
149
150
151    #stores log likelihoods
152    log_lik <- numeric(n_iter)
153
154    #Initialise Log-Likelihood and Objective Vectors - if there are informative priors,
            log likelihood of prior is also defined.
155
156    if(!(all(shape0==1) & all(rate0==0) )){
157      #   print("Non-flat prior input")
158      log_prior <- log_lik
159    }
160
161
162
163    objective <- log_lik
164    conv <- 0
165    l <- 1
166
167    best_objective <- -20000
168    best_p <- p
169    best_betaT <- betaT
170    best_z_hat <- NULL
171    best_piks <- NULL
172
173    # intialisation before main EM:
174    init <- ParV0(pi_inv   = pi_inv,
175                  G        = G,
176                  K        = K,
177                  n_rank   = n_rank,
178                  u_bin    = u_bin,
179                  shape0   = shape0,
180                  rate0    = rate0,
181                  rho      = rho,
182                  iterations = 500)
183    p    <- init$p
184    z    <- init$z
185    piV0 <- init$piV0
186    piks  <- upPis(X = X, betaT = betaT)
187    print(p)
188    print(betaT)
189    while(l<=n_iter){
190      if (l %% 50 == 0) {
191        cat("Iteration:", l, "\n")
192        print(p)
193        print(betaT)
194      }
195
196
197
198      # E step
199      z_hat <- Estep_z(p=p,ref_order=rho,piks= piks,pi_inv=pi_inv)
200      print(z_hat)
201      Dot <- UpDot(X=X, betaT = betaT)
202      print(Dot)
203      C <- upC(Dot=Dot)
204      print(C)
205      psi<-upPsi(Dot = Dot, C = C)
206      print(psi)
207      omega_store <-Estep_omega(psi=psi, z_hat)
208      print(omega_store)
209      kappas <- UpKappas(z_hat, omega_store, C)
```

```
210      print(kappas)

211

212      # M step - mixture weights
213      for(g in 2:G){

214

215        omega_g <- omega_store[,g]
216        omega_g <- omega_g * z_hat[,g]
217        kappa_g <- kappas[,g]
218        Omega_g <- diag(omega_g)
219        betaT[g,] <- upBhet(X, Omega_g, kappa_g, PriorCovariance)
220      }

221

222      betaT[1,] <- 0

223

224

225      # M step - support parameters
226      p <- UpPhetpartial(p=p,pi_inv=pi_inv,z_hat=z_hat,shape0=shape0,
227                         rate0=rate0,n_rank=n_rank,u_bin=u_bin)

228

229      if(any(is.na(p))){
230        print("==>  PROBLEM WITH *p* update")
231        print(p)
232      }

233

234      # calculates log likelihood of observed data given the current parameter
         estimates
235      log_lik[l] <- loglikPLMIXoE(p=p,piks=piks,pi_inv=pi_inv)
236      if (is.na(log_lik[l])) {
237        p[p < 1e-12] <- 1e-12
238        piks[piks < 1e-12] <- 1e-12
239        log_lik[l] <- loglikPLMIXoE(p=p,piks=piks,pi_inv=pi_inv)
240      }
241      if(is.na(log_lik[l])){

242

243        threshold <- -17
244        while(is.na(log_lik[l]) & threshold<(-3)){
245          p[p<=(10^threshold)] <- 10^threshold
246          threshold  <-  threshold+1
247          log_lik[l] <- loglikPLMIXoE(p=p,piks=piks,pi_inv=pi_inv)
248      print(log_lik[l])
249        }
250      }

251

252

253      betaT_LL <- betaT[-1, , drop = FALSE]
254      log_prior_beta <- sum(dnorm(as.vector(betaT_LL), mean=0, sd=10, log=TRUE))
255      if(!(all(shape0==1) & all(rate0==0) & all(shape0==1))){
256        log_prior[l] <- sum(dgamma(p, shape=shape0, rate=rate0, log=TRUE)) + log_prior_
         beta
257        objective[l] <- log_lik[l]+log_prior[l]

258

259      }else{
260        objective[l] <- log_lik[l]
261      }

262

263

264      # Save the best parameters if the current objective is better
265      if (!is.na(objective[l]) && (objective[l] > best_objective)) {
266        best_objective <- objective[l]
267        best_p <- p
268        best_betaT <- betaT
269        best_z_hat <- z_hat
270        best_piks <- piks
271      }

272

273      # convergence check
274      if(l >= 2){
275        if(is.na(objective[l-1]) || is.na(objective[l])){
```

```
276          warning("Encountered NA in objective function values; skipping convergence
       check for this iteration.")
277        } else if(abs(objective[l-1]) < eps){
278
279          if(abs(objective[l] - objective[l-1]) < eps){
280            conv <- 1
281            l <- n_iter + 1
282          }
283        } else {
284          # Normal case
285          if((objective[l] - objective[l-1]) / abs(objective[l-1]) < eps){
286            conv <- 1
287            l <- n_iter + 1
288          }
289        }
290      }
291      l <- l + 1
292      piks <- upPis(X=X, betaT=betaT)
293
294    }
295
296    p <- best_p
297    betaT <- best_betaT
298    z_hat <- best_z_hat
299    piks <- best_piks
300
301    #Post Processing After EM Loop
302    P_map=p/rowSums(p)
303    dimnames(P_map)=list(paste0("g_",1:G),paste0("p_",1:K))
304
305    names(piks) <- paste0("w_", M:G)
306
307
308    #Extract Final Log Likelihood and Objective Values
309    log_lik <- log_lik[!(is.na(log_lik))]
310    max_log_lik <- max(log_lik)
311
312    objective <- objective[!(is.na(objective))]
313    max_objective <- max(objective)
314
315    #BIC / DICM
316    if(all(shape0==1) & all(rate0==0) & all(shape0==1)){
317      bic <- bicPLMIX(max_log_lik=max_log_lik,pi_inv=pi_inv,
318                      G=G,ref_known=ref_known,
319                      ref_vary=ref_vary)$bic
320    }else{
321      bic <- NULL
322    }
323
324
325    if(plot_objective){
326      plot(objective,ylab="Log-joint distribution",xlab="Iteration",
327           main=paste("MAP estimation for PL mixture with",G,"components"),type="l")
328    }
329
330
331 #list of outputs
332    out=list(W_map=piks,P_map=P_map,z_hat=z_hat,class_map=apply(z_hat,1,which.max),
333             log_lik=log_lik,objective=objective,max_objective=max_objective,bic=bic,
       conv=conv,call=cl, betaT = betaT)
334    class(out)="mpPLMIX"
335    return(out)
336
337 }
```

61