Efficient and Accurate Learning of Mixtures of Plackett-Luce Models

Duc Nguyen[1], **Anderson Y. Zhang**[2]

[1] Depart of Computer and Information Science, University of Pennsylvania
[2] Department of Statistics and Data Science, University of Pennsylvania
mdnguyen@seas.upenn.edu, ayz@wharton.upenn.edu

Abstract

Mixture models of Plackett-Luce (PL), one of the most fundamental ranking models, are an active research area of both theoretical and practical significance. Most previously proposed parameter estimation algorithms instantiate the EM algorithm, often with random initialization. However, such an initialization scheme may not yield a good initial estimate and the algorithms require multiple restarts, incurring a large time complexity. As for the EM procedure, while the E-step can be performed efficiently, maximizing the log-likelihood in the M-step is difficult due to the combinatorial nature of the PL likelihood function. Therefore, previous authors favor algorithms that maximize surrogate likelihood functions. However, the final estimate may deviate from the true maximum likelihood estimate as a consequence. In this paper, we address these known limitations. We propose an initialization algorithm that can provide a provably accurate initial estimate and an EM algorithm that maximizes the true log-likelihood function efficiently. Experiments on both synthetic and real datasets show that our algorithm is competitive in terms of accuracy and speed to baseline algorithms, especially on datasets with a large number of items.

Introduction

Learning to rank is an active area of research with wide-ranging applications in recommendation systems, information retrieval, crowdsourcing and the social sciences. The Plackett-Luce (PL) model (Plackett 1975; Luce 1959) is one of the most fundamental ranking models. In a universe of n items, the PL model posits that item i has a latent *utility* $\theta_i^* \in \mathbb{R}$. The probability of observing a full ranking π given by the user (most preferred item first) is given as

$$\mathbb{P}^{PL}(\pi = [\pi_1, .., \pi_n] \,|\theta^*) = \prod_{i=1}^{n-1} \frac{\exp\left(\theta^*_{\pi_i}\right)}{\sum_{j=i}^{n} \exp\left(\theta^*_{\pi_j}\right)}. \quad (1)$$

The maximum likelihood estimate (MLE) can be obtained using iterative algorithms such as the Minorize-Maximize (MM) algorithm of Hunter (2004) and enjoys favorable theoretical properties (Hajek, Oh, and Xu 2014). In recent years, an algorithm known as Luce spectral ranking (LSR) (Maystre and Grossglauser 2015) has become the method of choice for maximum likelihood inference for PL models. LSR outputs the MLE just like MM but is often much faster.

The PL model is closely connected to the Bradley-Terry-Luce (BTL) model (Luce 1959) for *pairwise comparisons*. For two items $i \neq j$, the probability that i is ranked ahead of j *in a ranking* is equal to the probability that i beats j in a *pairwise comparison* under the BTL model. That is,

$$\mathbb{P}^{PL}(\pi(i) < \pi(j)) = \mathbb{P}^{BTL}_{ij} = \frac{1}{1 + \exp\left(-(\theta^*_i - \theta^*_j)\right)}, \quad (2)$$

where $\pi(i)$ is the position of item i in ranking π.

The classical PL model assumes that there is a universal preference ordering of the items according to their utilities. However, in practice, there might be multiple subpopulations of users with different preference profiles which cannot be fully captured by a single PL model. In such settings, a mixture of PL models is a more appropriate modeling assumption.

Problem Descriptions. Consider a mixture model with K components and n items for some constant K. Let $\beta^* = [\beta^*_1, \ldots, \beta^*_K]^\top$, ${\beta^*}^\top \mathbf{1} = 1$ denote the mixing distribution. For component $k \in [K]$ (where $[a]$ denotes $[1, \ldots, a]$), the utility parameters for the items are

$$\theta^{*k} = [\theta^{*k}_1, \ldots, \theta^{*k}_n]^\top \in \mathbb{R}^n.$$

Let $\boldsymbol{\theta}^* = [\theta^{*1}, \ldots, \theta^{*K}] \in \mathbb{R}^{n \times K}$ denote the concatenation of the K sets of parameters. A *ranking dataset* Π is a collection of full rankings.

Consider the following generative model for a ranking dataset of size m. For $l \in [m]$, let $z^*_l \in [K]$ denote the mixture component membership where $\mathbb{P}(z^*_l = k) = \beta^*_k$. Then a permutation π_l is drawn from the PL distribution parametrized by θ^{*z_l}. That is,

$$\mathbb{P}^{PL}(\pi_l = [\pi_{l,1}, \ldots, \pi_{l,n}] \,|\, z^*_l, \boldsymbol{\theta}^*) = \prod_{i=1}^{n-1} \frac{\exp\left(\theta^{*z^*_l}_{\pi_{l,i}}\right)}{\sum_{j=i}^{n} \exp\left(\theta^{*z^*_l}_{\pi_{l,j}}\right)}, \quad (3)$$

where $\pi_{l,i}$ denote the i-th item in permutation π_l. The reader may recognize two identifiability issues here. The first is parameter translation. For each component, the distributions parametrized by θ^{*k} and $\theta^{*k} + c \cdot \mathbf{1}_n$ are the same for any $c \in \mathbb{R}$. The second is mixture components (columns of $\boldsymbol{\theta}^*$)

relabeling. To account for these issues, we consider the following error metric.

$$\text{dist}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) := \min_{R \in \mathcal{O}^{K \times K}} \|N(\boldsymbol{\theta})R - N(\boldsymbol{\theta}^*)\|_F \,, \quad (4)$$

where $\mathcal{O}^{K \times K}$ is the set of all permutation matrices (Strang et al. 1993, Chapter 2) of size $K \times K$ and N is the normalization operator (i.e., $N(\boldsymbol{\theta})_{\cdot k} = \theta^k - \frac{1}{n}(\theta^k)^\top \mathbf{1}_n$).

Prior Works. Generalizing the PL model to mixtures adds a layer of complexity to the inference problem. In general, the likelihood function is non-convex in the model parameters. Most previously proposed algorithms instantiate the EM algorithm (Dempster, Laird, and Rubin 1977). As a general recipe, an EM algorithm is initialized with some parameter $\boldsymbol{\theta}^{(0)}$ (e.g., using random initialization). It then repeats the following two steps for $t = 1, 2, \ldots$ until convergence.

The E-step computes the posterior class probability conditioned on the current estimate:

$$q_l^k = \mathbb{P}(z_l^* = k \,|\, \pi_l, \boldsymbol{\theta}^{(t-1)}) \propto \beta_k \cdot \mathbb{P}^{PL}(\pi_l \,|\, z_l^* = k, \boldsymbol{\theta}^{(t-1)}) \quad (5)$$

for $l \in [m], k \in [K]$ where $\mathbb{P}^{PL}$ is given in Equation (3) and β the prior class probability. Thanks to the closed form of the PL likelihood function, the E-step can be done efficiently. The M-step obtains the next estimate $\boldsymbol{\theta}^{(t)}$ by maximizing the joint log-likelihood function which decomposes into K *weighted log-likelihood functions*. Namely,

$$\boldsymbol{\theta}^{(t)} = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{K} \left(\sum_{l=1}^{m} q_l^k \log \mathbb{P}^{PL}(\pi_l, z_l^* = k \,|\, \boldsymbol{\theta}) \right). \quad (6)$$

Due to the combinatorial nature of the PL likelihood function, the derivative of the log likelihoood function has a complicated form. As a result, maximizing the (weighted) log-likelihood via gradient-based algorithms quickly becomes inefficient as n grows.

The first practical approach towards solving the M-step uses the Minorize-Maximize algorithm of Hunter (2004), yielding the EMM algorithm of Gormley and Murphy (2008). While guaranteed to solve the M-step, it has been observed that the MM subroutine converges slowly even for datasets with a moderate number of items (e.g., Figure 2). Motivated by practical concerns, researchers have developed *pseudo-likelihood* estimators that optimize, instead of the true log-likelihood function, *alternative objective functions*. Two such algorithms are the Generalized Method of Moments (GMM) of Azari Soufiani et al. (2013) and Composite Marginal Likelihood (CML) of Zhao and Xia (2018). It has been observed experimentally that GMM is considerably faster than MM and CML is even faster than GMM with comparable accuracy. Besides maximum likelihood (ML) inference methods, previous authors have also proposed Bayesian inference algorithms (Guiver and Snelson 2009; Mollica and Tardella 2017). In this paper, we focus primarily on ML algorithms but include additional experiments with Bayesian methods in the supplementary materials.

Using GMM and CML to solve the M-step gives us the EM-GMM algorithm (Zhao, Villamil, and Xia 2018) and the EM-CML algorithm (Zhao, Liu, and Xia 2020), respectively. The only non-EM algorithm for learning PL mixtures that we are aware of is a GMM-based algorithm proposed in Zhao, Piech, and Xia (2016); Zhao and Xia (2019). However, the construction of the algorithm is quite ad-hoc and the authors did not show extension of the algorithm to more than 2 mixture components. In addition, previous authors primarily restrict their experiments to datasets with a small number of items such as the SUSHI datasets (Kamishima 2003) with $n = 10$. It is unknown how the previous methods perform when n is large. Recent works have also studied PL mixtures learning with features and partial rankings (Tkachenko and Lauw 2016; Liu et al. 2019). While we include possible extensions of our algorithm in the supplementary materials, our main focus in this paper is an *improved algorithm for the classical setting*.

Our Contributions. We propose a new EM algorithm for learning mixtures of PL models that

- Has a provably accurate initialization procedure with a finite sample error guarantee, the first of its kind in the literature;
- Efficiently maximizes the weighted log-likelihood function in the M-step without using a surrogate likelihood or objective function, thus returning the true maximum likelihood estimate;
- Performs competitively with the previously proposed algorithms in terms of accuracy and speed, and is scalable to datasets with $n \geq 100$.

The Spectral EM Algorithm

In this section, we present our algorithmic contributions. Section describes the spectral initialization algorithm and Section describes the EM refinement procedure.

Spectral Initialization

The initialization for our algorithm is delegated to spectral clustering (Algorithm 1) and a least squares minimization algorithm (Algorithm 2). To apply spectral clustering, we first embed each ranking π_l into a 'pairwise vector' – $X_l \in \{0,1\}^{\binom{n}{2}}$ where each entry corresponds to a pair of items. As an overload of notation, we use $d = (d_1, d_2)$ where $d_1 < d_2$ to denote the entry corresponding to the pair (d_1, d_2). Define

$$X_{l,d}(\pi) = \begin{cases} 1 & \text{if } \pi_l(d_1) < \pi_l(d_2) \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

Let $X \in \mathbb{R}^{m \times \binom{n}{2}}$ denote the concatenation of the embeddings of m rankings in dataset Π. Given a target number of components K, Algorithm 1 can then be applied to the rows of X to obtain K clusters, $\{\hat{C}^k\}_{k=1}^K \subseteq [m]$.

For each cluster of rankings $\hat{C}^k$, we estimate the preference probability for a pair (i, j) as

$$\hat{P}_{ij}^k = \frac{1}{|\hat{C}^k|} \sum_{l \in \hat{C}^k} \mathbf{1}[\pi_l(i) < \pi_l(j)] \,. \quad (8)$$

From the preference probability estimates for all pairs, Algorithm 2 recovers the utility parameter $\hat{\theta}^k$. It applies the

Algorithm 1 Spectral Clustering with Adaptive Dimension Reduction

Input: Dataset $\Pi = \{\pi_1, \ldots, \pi_m\}$, number of mixture components K and threshold T.

Output: K clusters of rankings.

1: Embed the rankings as the rows of a matrix $X \in \{0,1\}^{m \times \binom{n}{2}}$ according to Equation (7).
2: Perform SVD: $X = USV^\top$, where the singular values are arranged from largest to smallest.
3: Let $\hat{r}$ be largest index in $[K]$ such that the difference between the successive singular values is greater than T, i.e., $\hat{r} = \max\{a \in [K] : S_{aa} - S_{(a+1)(a+1)} \geq T\}$.
4: Run k-means on the rows of $XV_{1:\hat{r}}$ with K clusters:

$$\left(\hat{z}, \{\hat{c}_k\}_{k=1}^K\right) = \underset{\substack{z \in \{1,\ldots,K\}^m \\ \{c_k\} \in \mathbb{R}^{\hat{r}}}}{\arg\min} \sum_{l=1}^{m} \|V_{1:\hat{r}}^\top X_l - c_{z_l}\|_2^2 .$$

5: Return clusters $\hat{C}^k = \{l \in [m] : \hat{z}_l = k\}$ for $k \in [K]$.

Algorithm 2 Least Squares Parameter Estimation

Input: Pairwise preference matrix $\hat{P} \in \mathbb{R}^{n \times n}$.

Output: Normalized parameter estimate $\hat{\theta}$.

1: Solve the least squares optimization problem

$$\hat{\theta} = \arg \max_{\theta : \theta^\top \mathbf{1} = 0} \sum_{i \neq j} (\hat{\phi}_{ij} - (\theta_i - \theta_j))^2 ,$$

where $\hat{\phi}_{ij} = \ln(\hat{P}_{ij}/(1 - \hat{P}_{ij}))$.

logit function on the pairwise probabilities and solves a constrained least squares minimization problem, which can be efficiently done using off-the-shelf solvers (Virtanen et al. 2020). Algorithm 3 summarizes the spectral initialization algorithm.

Remarks. The application of spectral clustering to mixtures of PL models has also appeared in a manuscript by Shah and Song (2018). There, the authors apply the classical spectral clustering algorithm – clustering the rows of $XV_{1:K}$ – and their analysis requires a spectral gap condition which is hard to verify. We use spectral clustering with adaptive dimension reduction and our analysis does not require any spectral gap condition (Zhang and Zhou 2022). Furthermore, we focus on *parameter estimation* while Shah and Song (2018) only focus on clustering, resulting in different theoretical guarantees. The choice of threshold T in Algorithm 3 is to satisfy a mild technical condition in the analysis of spectral clustering. In our experiments, the performance of the EM algorithm does not seem to critically depend on this threshold.

Intuition behind Algorithm 2. Recall the connection between the PL model and the BTL model in Equation (2). Suppose we observe a large sample drawn from a single PL distribution. Then $\hat{P}_{ij} \approx P^*_{ij} = 1/\left(\exp\left(-(\theta^*_i - \theta^*_j)\right)\right)$ and $\hat{\phi}_{ij} = \ln(\hat{P}_{ij}/1 - \hat{P}_{ij}) \approx \theta^*_i - \theta^*_j$. Solving the least squares optimization problem recovers $\hat{\theta} \approx \theta^*$. In the mixture setting, if the estimates $\hat{P}^k$'s are accurate, we obtain good parameter estimates (e.g., Theorem 1). Rajkumar and Agarwal (2016) apply a similar idea in their algorithm for *ranking from comparisons of $O(n \log n)$ pairs under a single BTL model*. They first apply the logit function on the pairwise preference probabilities, followed by a low rank matrix completion algorithm (Keshavan, Montanari, and Oh 2009). Their algorithm *produces a ranking*. On the other hand, our goal is mixture learning and the resulting theoretical analysis is different.

Algorithm 3 Spectral Initialization

Input: Dataset $\Pi = \{\pi_1, \ldots, \pi_m\}$, number of mixture components K.

Output: Parameter estimates for K mixture components $\hat{\boldsymbol{\theta}} = [\hat{\theta}^1, \ldots, \hat{\theta}^K] \in \mathbb{R}^{n \times K}$.

1: Run Algorithm 1 on Π with $T = \sqrt{n}\sqrt{m+n}\sqrt{\log n}$ to obtain K clusters $\hat{C}^1, \ldots, \hat{C}^K$.
2: Estimate the pairwise preference probabilities $\hat{P}^k_{ij}$ per Equation (8) for each cluster.
3: Run Algorithm 2 on $\{\hat{P}^k\}_{k=1}^K$ and return the parameter estimates for K mixture components.

Iterative Refinement via EM

The Weighted LSR Algorithm. As noted before, we wish to maximize the weighted log-likelihood (6) efficiently. Towards this goal, we generalize the Luce spectral ranking (LSR) algorithm (Maystre and Grossglauser 2015) to incorporates sample weights. The original LSR algorithm produces the MLE. Our generalized algorithm outputs the weighted MLE (see Theorem 2).

The intuition behind LSR is an interpretation of the *PL ranking generative process as a sequence of choices* (Plackett 1975). Given a ranking π_l, define its *choice breaking* as

$$\mathcal{B}_{\pi_l} = \{(\pi_{l,1}, \{\pi_{l,1}, \ldots, \pi_{l,n}\}, l), (\pi_{l,2}, \{\pi_{l,2}, \ldots, \pi_{l,n}\}, l) \ldots, (\pi_{l,n-1}, \{\pi_{l,n-1}, \pi_{l,n}\}, l)\} .$$

Each tuple $(i, A, l) \in \mathcal{B}(\pi_l)$ is a *choice enumeration* of the ranking π_l. Given a ranking dataset $\Pi = \{\pi_1, \ldots, \pi_m\}$, define the *choice breaking* of Π as the union of all ranking-level choice breakings:

$$\mathcal{B}_\Pi = \mathcal{B}_{\pi_1} \cup \ldots \cup \mathcal{B}_{\pi_m} . \tag{9}$$

Note that $|\mathcal{B}_\Pi| = m(n-1)$. When the dataset Π is clear from context, we simply use $\mathcal{B}$ to denote the dataset-level choice breaking.

We now introduce sample weights. Firstly, define the 'weight' of a choice breaking $\mathcal{B}$ with weight vector w and parameter $\theta \in \mathbb{R}^n$ as

$$\gamma(\mathcal{B}, w, \theta) = w^\top \left(\frac{1}{\sum_{j \in A} e^{\theta_j}} \right)_{(i,A,l) \in \mathcal{B}} , \tag{10}$$

where $w \in \mathbb{R}_+^{m(n-1)}$ is an arbitrary weight vector;

$\left(\frac{1}{\sum_{j\in A} e^{\theta_j}}\right)_{(i,A,l)\in\mathcal{B}}$ is also vector in $\mathbb{R}_+^{m(n-1)}$ where each entry corresponds to a choice enumeration (i, A, l).

The reader may recognize that the weight vector w has the same size as the choice breaking while sample weights are often given at the ranking level – each ranking π_l is assigned a weight q_l for $l \in [m]$ as in (6). Given sample weights $q = (q_1, \ldots, q_m)$, one simply sets

$$w = [\underbrace{q_1, \ldots, q_1}_{n-1 \text{ terms}}, \underbrace{q_2, \ldots, q_2}_{n-1 \text{ terms}}, \ldots, \underbrace{q_m, \ldots, q_m}_{n-1 \text{ terms}}]^\top . \quad (11)$$

Given a choice breaking $\mathcal{B}$ and items i, j, define the set of choice enumerations where i 'beats' j as

$$\mathcal{B}_{i\succ j} = \{(i, A, l) \in \mathcal{B} \,:\, j \in A\} .$$

As a shorthand notation, for a weight vector w corresponding to choice breaking $\mathcal{B}$, define $w_{j\succ i}$ as the sub-vector of w corresponding to $\mathcal{B}_{i\succ j}$.

Similarly to the original LSR algorithm, we construct a Markov chain (MC) and recover PL parameters from its stationary distribution. This MC has n states. Given choice breaking $\mathcal{B}$, weight vector w and parameter θ, the pairwise transition probabilities of M are given as

$$M_{ij} = \begin{cases} \frac{1}{d} \cdot \gamma(\mathcal{B}_{j\succ i}, w_{j\succ i}, \theta) & \text{if } i \neq j \\ 1 - \frac{1}{d} \cdot \sum_{k\neq i} \gamma(\mathcal{B}_{k\succ i}, w_{k\succ i}, \theta) & \text{if } i = j \end{cases} , \quad (12)$$

where d is a sufficiently large normalization constant such that M does not contain any negative entries. Intuitively, M_{ij} is proportional to the sum of the weights of all choice enumerations where j 'beats' i.

Algorithm 4 summarizes the weighted LSR algorithm. It repeatedly constructs a Markov chain based on the current estimate, computes its stationary distribution and recovers the next estimate until convergence. When sample weights are not given, the weighted LSR algorithm reduces to the original LSR algorithm.

The EM-LSR Algorithm. In the E-step, we compute the posterior class probabilities $q^k \in \mathbb{R}^m, k \in [K]$. The M step consists of K maximization problem as shown in Equation (6). These can be solved in parallel by running Algorithm 4 on Π using q^k as sample weights for $k \in [K]$. Algorithm 5 summarizes the overall algorithm.

In another EM-based approach for learning PL mixtures, Liu et al. (2019) use the *unweighted LSR* algorithm. There, the E-step remains the same. The key differences lie in initialization (they use random initialization) and in the M-step. Our algorithm maximizes the weighted log-likelihood via weighted LSR and is therefore an exact EM algorithm. On the other hand, Liu et al. use the posterior class probabilities to perform a random clustering of the rankings and then run unweighted LSR on each cluster, making their algorithm an *inexact* EM algorithm. From additional experiments in the supplementary materials, one can observe that the stochastic M-step actually leads to *worse estimates without a significant reduction in inference time*.

Algorithm 4 Weighted Luce Spectral Ranking

Input: Dataset $\Pi = \{\pi_1, \ldots, \pi_m\}$, (optional) weight vector $q \in \mathbb{R}_+^m$ and (optional) initial estimate $\hat{\theta}^{(0)} \in \mathbb{R}^n$.

Output: Normalized estimate of the item parameters $\hat{\theta} \in \mathbb{R}^n$.

1: Obtain choice breaking $\mathcal{B}$ from Π per Equation (9).
2: If the weight vector q is not given, set $q = \mathbf{1}_m$.
3: Construct w from q per Equation (11).
4: If the initial estimate is not given, set $\hat{\theta}^{(0)} = \mathbf{0}_n$.
5: For $t = 1, \ldots$ until convergence
  5.1: Construct a Markov chain M with pairwise transition probability per Equation (12) from choice breaking $\mathcal{B}$, weight vector w and parameter $\hat{\theta}^{(t-1)}$.
  5.2: Compute the stationary distribution of M (e.g., via power iteration), p and return the normalized estimate $\hat{\theta}^{(t)} = \log(p) - \left(\frac{1}{n}\sum_{i=1}^n \log(p)\right) \cdot \mathbf{1}_n$

Algorithm 5 Spectral EM (EM-LSR)

Input: Dataset $\Pi = \{\pi_1, \ldots, \pi_m\}$, number of components K, prior distribution β, (optional) initial estimate $\hat{\boldsymbol{\theta}}^{(0)} \in \mathbb{R}^{n\times K}$.

Output: Normalized estimate $\hat{\boldsymbol{\theta}} = [\hat{\theta}^1, \ldots, \hat{\theta}^K]$.

1: If $\hat{\boldsymbol{\theta}}^{(0)}$ is not given, run Algorithm 3 on Π with K mixture components and set $\hat{\boldsymbol{\theta}}^{(0)}$ to the output.
2: For $t = 1, 2, \ldots$ until convergence
  2.1: E-step – Compute the class posterior probabilities $q_l^k = p(z_l^* = k|\pi_l, \hat{\boldsymbol{\theta}}^{(t-1)})$ for $l \in [m], k \in [K]$.
  2.2: M-step – Estimate $\hat{\theta}^{k(t)}$ by running Algorithm 4 on Π with sample weight vector $q^k = [q_1^k, \ldots, q_m^k]$ and initial estimate $\hat{\theta}^{k(t-1)}$ for $k \in [K]$.

Theoretical Analysis

In this section, we study the theoretical properties of EM-LSR. Specifically we present the finite sample error guarantee for the spectral initialization algorithm and study the M-step of not only our EM-LSR algorithm but also related EM algorithms.

Spectral Initialization

Central to the analysis of the spectral initialization algorithm is the accuracy of spectral clustering (Algorithm 1). Our analysis starts from the fact that, under the pairwise representation in Equation (7), the PL distribution exhibits *sub-gaussian characteristics* (Vershynin 2018; Shah and Song 2018). The detailed descriptions of these characteristics are not immediately important to our discussions so we refer the interested reader to the supplementary materials. However, we emphasize that these characteristics also appear in a broad class of ranking models known as *random utility models* (RUMs) that subsume the PL model. The spectral clustering algorithm is model-agnostic. It can be applied to mixtures of sub-gaussian distributions and enjoys high clustering accuracy if the signal-to-noise ratio (SNR) is high. We also show how, by changing

the mapping function used in Algorithm 2, we can perform parameter estimation for a general RUM, not just PL. Thanks to this flexibility, Algorithm 3 can be a useful tool for learning mixtures of general RUMs.

We now consider an expressive generative model for mixtures of K PLs where Algorithm 3 produces a provably accurate estimate. The generative model assumes that for all mixture components, only the utilities of the first L items are different while the those of the remaining $n-L$ items are the same. This model reflects the phenomenon where users from different sub-populations differ in their preference among a few items while the remaining items are essentially interchangeable. Intuitively, one would expect that when L is small, so is the difference between the subpopulations and it is harder to separate the rankings into the correct clusters. On the other hand, when L is large, the difference among the subpopulations is large and it is easier to separate the clusters. The following theorem captures this intuition.

Theorem 1. *Consider a mixture of K Plackett-Luce models with uniform mixing probabilities. Suppose that $\theta_i^{*k} = 0 \; \forall i \in [L+1:n]$ and $\theta_{1:L}^{*k} \sim \mathbb{N}(0, I_L)$ for $k \in [K]$. Fix a constant $\alpha > 0$. There exist constants c, c_1, C_1, C_2, D such that if $m \geq c \max\{K^4, Kn\}$ then the output $\hat{\boldsymbol{\theta}}$ of Algorithm 3 satisfies the following. If $L \geq c_1 \exp\left(C_1 \sqrt{\log n}\right)$, then*

$$\textit{dist}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^*) = O\left(\exp\left(D\sqrt{\log n}\right)\left(\sqrt{\frac{K^2 n \log n}{m}} + \frac{\sqrt{Kn}}{e^{L^{0.99}}}\right)\right)$$

with probability $1 - O(\frac{K}{n^8}) - O(K^2 n^2 \exp\left(-L^{0.99}\right))$. If $L \geq C_2 n^\alpha$ and assuming that $n = \omega(\log m)$, then

$$\textit{dist}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^*) = O\left(\exp\left(D\sqrt{\log n}\right)\sqrt{\frac{K^2 n \log n}{m}}\right)$$

with probability $1 - O(\frac{K}{n^8}) - O(K^2 n^2 \exp\left(-n^\alpha\right))$.

The first error bound is a sum of two terms. The first is the estimation error incurred by Algorithm 2 which diminishes with increasing m. The second comes from the clustering error incurred by Algorithm 1 and is controlled by the SNR of the generative model. One can also check that $\exp\left(\sqrt{\log n}\right) = o(n^\alpha)$ for any $\alpha > 0$ and $\exp\left(\sqrt{\log n}\right) = \omega(\log n)$. When $L \approx \exp\left(O(\sqrt{\log n})\right)$ (low SNR), there is significant clustering error and the second term scales approximately as $O(\frac{\sqrt{n}}{e^L}) = O\left(\frac{1}{\text{poly}(n)}\right)$. Hence, Algorithm 3 converges to within a small radius around $\boldsymbol{\theta}^*$ given a sufficiently large m. However, when L is polynomial in n (high SNR), estimation error dominates clustering error, giving us the second error bound which diminishes with sample size m. In this regime, the spectral initialization algorithm works well as a *standalone mixture learning algorithm*. Note that this guarantee holds even for a small $\alpha > 0$, when the fraction of 'informative' items diminishes: $L/n = o(1)$. Our proposed generative model is new and could be a useful analysis framework for future works. To the best of our knowledge, the finite sample error bounds are also the *first of their kind in the literature*.

Iterative Refinement via EM

Accuracy of the M-step. The following theorem generalizes Theorem 1 of Maystre and Grossglauser (2015).

Theorem 2. *The output of weighted LSR (Algorithm 4) is the maximum weighted log-likelihood estimate:*

$$\theta_q^{MLE} := \arg\max_\theta \sum_{l=1}^m q_l \cdot \log \mathbb{P}^{PL}(\pi_l, z_l \,|\, \theta)\,.$$

As noted before, the EMM algorithm is an alternative approach that exactly solves the M-step using the (weighted) MM algorithm. In other words, *assuming perfect numerical precision and the same initialization*, EMM and EM-LSR will produce the same final estimate. However, our EM-LSR algorithm is often much faster than EMM (e.g., Figure 1).

Convergence of EM. It is well known that the EM algorithm converges to a stationary point (Wu 1983). There is, unfortunately, no guarantee how close such a point is to the global optimum. However, assuming correct model specification and that the initial estimate falls within a neighborhood around θ^* which satisfies certain high SNR conditions, the EM algorithm will converge to θ^* (Wang et al. 2015; Wu et al. 2016; Balakrishnan, Wainwright, and Yu 2017). The area around θ^* where this desirable behaviour occurs is referred to as the *basin of attraction*. We leave the detailed characterization of the basin of attraction as a subject of future studies.

True Likelihood versus Surrogate Likelihood. For two other commonly used EM algorithms in the literature – EM-CML and EM-GMM – previous authors use random initialization. On the other hand, ours uses spectral initialization. However, initialization is not the only differentiating characteristic of our algorithm. In fact, our algorithm, EM-CML and EM-GMM are *fundamentally different EM-based algorithms*. To see why, one needs to inspect the objective function of the M-step. Suppose that all three algorithms are initialized at some $\hat{\boldsymbol{\theta}}^{(0)}$. Let $\{q_l^k\}_{l\in[m]}^{k\in[K]}$ denote the posterior class probabilities conditioned on $\hat{\boldsymbol{\theta}}^{(0)}$ per Equation (5).

In the first iteration, EM-LSR and EMM maximize the weighted log-likelihood.

$$\hat{\boldsymbol{\theta}}_{\text{LSR}}^{(1)} = \arg\max_{\boldsymbol{\theta}} \sum_{l=1}^m \sum_{k=1}^K \left[q_l^k \cdot \log \mathbb{P}^{PL}(\pi_l, z_l \,|\, \theta^k)\right].$$

On the other hand, EM-CML maximizes the *composite (surrogate) marginal likelihood*. $\hat{\boldsymbol{\theta}}_{\text{CML}}^{(1)} = \arg\max_{\boldsymbol{\theta}}$

$$\sum_{l=1}^m \sum_{k=1}^K \left[\sum_{\substack{i,j:\\ \pi_l(i)<\pi_l(j)}} q_l^k \log\left(\frac{1}{1+\exp\left(-(\theta_i^k - \theta_j^k)\right)}\right)\right].$$

Lastly, EM-GMM *minimizes* the following function.

$$\hat{\boldsymbol{\theta}}_{\text{GMM}}^{(1)} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^K \sum_{i\neq j} \left(\hat{F}_{ij}^k - \frac{1}{1+\exp\left(-(\theta_i^k - \theta_j^k)\right)}\right)^2,$$

where $\hat{F}_{ij}^k = \frac{\sum_{l=1}^m \mathbf{1}[\pi_l(i)<\pi_l(j)]\, q_l^k}{\sum_{l=1}^m q_l^k}$.

One can see that the objective functions are different and so are their solutions. Hence, even if we initialize all three algorithms with the same estimate, their trajectories will be different in general. While EM-LSR and EMM converges to the true MLE when initialized within the basin of attraction, this *may not be true for EM-GMM and EM-CML*. This difference is supported by our experiments, where even with the same initialization, the algorithms produce different final estimates.

Experiments

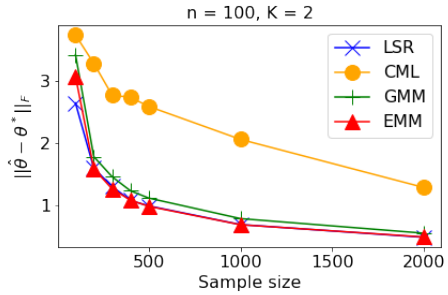
We compare our spectral EM algorithm to the following baselines: EMM, EM-GMM and EM-CML.

Synthetic Datasets. We simulate data from the generative model as described in Theorem 1. We set $n = 100$ and $L = 5$ while varying the number of mixture components K for different experiments. Figure 1 shows estimation error and total inference time against the sample size m, averaged over 25 trials. Spectral initialization consistently gives better initial estimates than both random initialization and GMM initialization (Zhao, Piech, and Xia 2016). To keep a fair comparison, we use spectral initialization for all algorithms. When K is small (e.g., Figures 1a and 1b) all four methods are quite accurate. When the number of mixture components are moderate (e.g., Figures 1c and 1d), the advantages that EM-LSR enjoys over the other methods become more apparent. While EMM becomes too inefficient for practical purposes, EM-LSR remains relatively efficient and produces more accurate estimates than both EM-CML and EM-GMM.

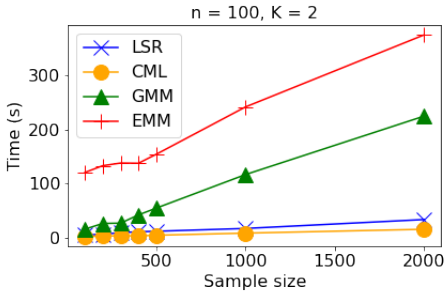
Real Datasets. We include commonly used datasets in previous works such as APA, Irish Elections (West, North, Meath) and SUSHI all with $n < 15$. We partition all the rankings with a 80-20 training-testing split; and the train rankings into 80% for inference and 20% for validation. K is chosen using Bayesian Information Criterion (Gelman, Hwang, and Vehtari 2014) on the validation set and the log-likelihood of the final model is evaluated using the test set. For these datasets, EM-LSR and EMM are the most accurate while EM-CML is the fastest, especially on datasets with a large m such as the Irish election datasets. To understand the relative speed between EM-LSR and EM-CML, note that the bottle neck in these EM algorithms is the M-step. The most time-consuming procedure in the M-step of EM-LSR is constructing the Markov chain in Algorithm 4 with time complexity $O(mn^2)$. For EM-CML, it is solving a constrained concave maximization problem via SLSQP (Virtanen et al. 2020) and may scale at least as $\Omega(n^3)$. [1]. Therefore, EM-CML tends to be faster for datasets with a small n and a large m. However, its inference time could grow significantly with n.

Indeed, the setting where EM-LSR outperforms the baselines is when n is large. We perform additional experiments on the ML-10M movie ratings datasets (Harper and Konstan 2015). To generate rankings, we first run a low rank matrix completion algorithm (Zitnik and Zupan 2012) on the user-

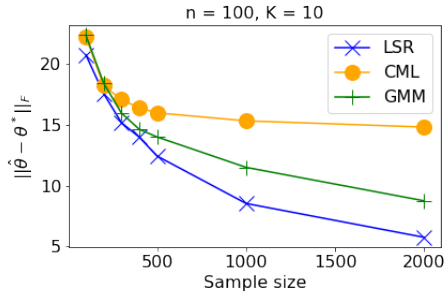
[1]SLSQP solves a sequence of quadratic optimization problems with n variables. Each solves a linear system with n variables and n equations and generally takes $O(n^3)$ (Strang et al. 1993).



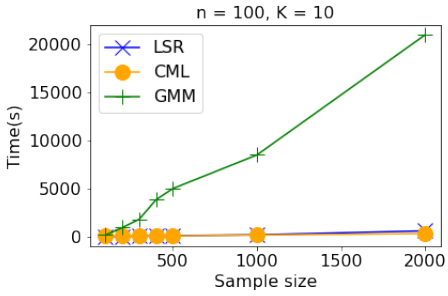
(a) For a small number of mixture components, all methods are quite accurate. As the theory implies, EMM and EM-LSR produce similar estimates.



(b) EM-LSR is comparatively efficient (figure shows total inference time).



(c) For a moderate number of mixture components, EM-LSR is the most accurate method (EMM not shown due to timeout).



(d) EM-LSR and EM-CML are the only two methods that are efficient for a moderate number of mixture components.

Figure 1. ℓ_2 error and inference time on synthetic datasets. EM-LSR is competitive in terms of accuracy and speed to the baseline algorithms.

item rating matrix to fill in the missing entries. We then select n movies from the set of all movies and the rankings are obtained from the completed matrix. Figure 2 shows the performance of the four methods on two versions of the ML-10M datasets with $n = 25$ and $n = 100$ given increasing training data up to 14k. In the supplementary materials (Nguyen and Zhang 2023), we also include additional experiments, strategies to extend EM-LSR to handle partial rankings with ties and comparisons to a Bayesian method (Mollica and Tardella 2017). Table 1 summarizes the experimental results on real datasets.

	Test log-likelihood (inference time)			
Dataset	LSR	CML	GMM	EMM
APA	-4.6 (600)	-4.6 (**33**)	-4.6 (2.2K)	**-4.6** (9.24K)
West	-11.9 (810)	-12.0 (**200**)	-11.9 (5.8K)	**-11.9** (25K)
Sushi	**-13.6** (746)	-14.0 (**24.6**)	-13.8 (489)	-13.8 (1.2K)
North	**-18.7** (1.5K)	-18.9 (**120**)	-18.7 (3.1K)	-18.8 (14K)
Meath	-23.6 (1.5K)	-23.9 (**497**)	-23.7 (30K)	**-23.6** (70K)
ML (25)	-49.2 (3.7K)	-50.1 (**2.5K**)	-49.8 (26K)	**-49.2** (63K)
ML (50)	**-131 (5.8K)**	-132 (6.8K)	-132 (125K)	NA
ML (100)	**-326 (12K)**	-330 (27K)	-332 (490K)	NA
ML (200)	**-787 (25K)**	-799 (81K)	NA	NA

Table 1. Test log-likelihood and inference time on real datasets. 'NA' denotes not available due to timeout. Best performances are in bold.

Conclusion

We have proposed an accurate and efficient algorithm for learning a mixture of Plackett-Luce models. For future works, we would like to consider other initialization methods such as the method of moments or tensor decomposition. Detailed characterization of the basin of attraction within which the EM algorithm converges to the true parameter is also a challenging open question. On a more practical note, incorporating the representation power of deep neural networks into our algorithm will further increase its utility for large scale recommendation systems applications.

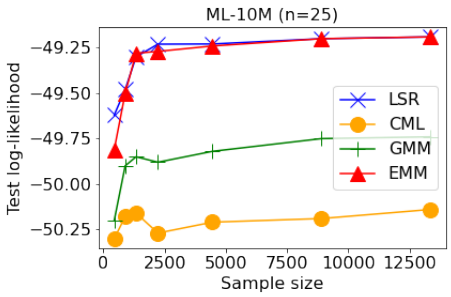
Acknowledgements

The authors thank the anonymous reviewers for their thoughtful suggestions and comments. The authors are supported by NSF Grant DMS-2112099. A.Z. acknowledges financial support from the Alfred H. Williams Faculty Scholar award.

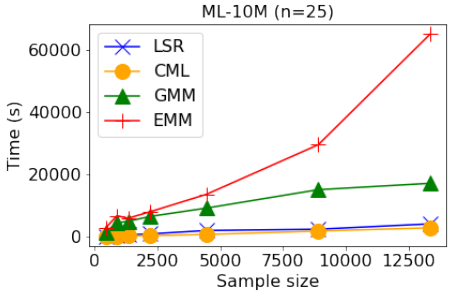
References

Azari Soufiani, H.; Chen, W.; Parkes, D. C.; and Xia, L. 2013. Generalized method-of-moments for rank aggregation. *Advances in Neural Information Processing Systems*, 26.

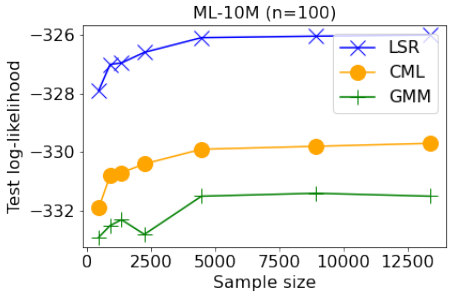
Balakrishnan, S.; Wainwright, M. J.; and Yu, B. 2017. Statistical guarantees for the EM algorithm: From population



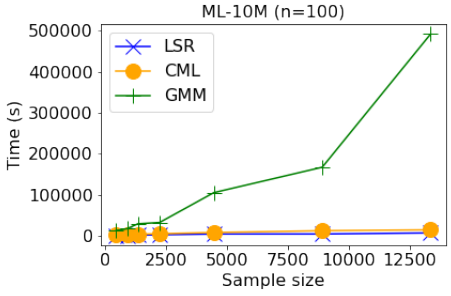
(a) With a small number of items, EMM and EM-LSR are more accurate.



(b) However, EM-LSR is also competitive in terms of efficiency.



(c) For a larger set of items, EM-LSR is the most accurate method (EMM not shown due to timeout).



(d) EM-LSR is comparatively scalable for larger datasets.

Figure 2. Test log-likelihood and inference time on ML-10M datasets. For larger datasets, EM-LSR is more accurate while being competitive in speed to the baseline algorithms.

to sample-based analysis. *The Annals of Statistics*, 45(1): 77–120.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22.

Gelman, A.; Hwang, J.; and Vehtari, A. 2014. Understanding predictive information criteria for Bayesian models. *Statistics and computing*, 24(6): 997–1016.

Gormley, I. C.; and Murphy, T. B. 2008. Exploring voting blocs within the Irish electorate: A mixture modeling approach. *Journal of the American Statistical Association*, 103(483): 1014–1027.

Guiver, J.; and Snelson, E. 2009. Bayesian inference for Plackett-Luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, 377–384.

Hajek, B.; Oh, S.; and Xu, J. 2014. Minimax-optimal inference from partial rankings. *Advances in Neural Information Processing Systems*, 27.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.

Hunter, D. R. 2004. MM algorithms for generalized Bradley-Terry models. *The annals of statistics*, 32(1): 384–406.

Kamishima, T. 2003. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 583–588.

Keshavan, R.; Montanari, A.; and Oh, S. 2009. Matrix completion from noisy entries. *Advances in neural information processing systems*, 22.

Liu, A.; Zhao, Z.; Liao, C.; Lu, P.; and Xia, L. 2019. Learning plackett-luce mixtures from partial preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4328–4335.

Luce, R. D. 1959. *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Maystre, L.; and Grossglauser, M. 2015. Fast and accurate inference of Plackett–Luce models. *Advances in neural information processing systems*, 28.

Mollica, C.; and Tardella, L. 2017. Bayesian Plackett–Luce mixture models for partially ranked data. *Psychometrika*, 82(2): 442–458.

Nguyen, D.; and Zhang, A. Y. 2023. Efficient and Accurate Learning of Mixtures of Plackett-Luce Models. *arXiv preprint arXiv:2302.05343*.

Plackett, R. L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2): 193–202.

Rajkumar, A.; and Agarwal, S. 2016. When can we rank well from comparisons of O (n\log (n)) non-actively chosen pairs? In *Conference on Learning Theory*, 1376–1401. PMLR.

Shah, D.; and Song, D. 2018. Learning RUMs: Reducing Mixture to Single Component via PCA. *arXiv preprint arXiv:1812.11917*.

Strang, G.; Strang, G.; Strang, G.; and Strang, G. 1993. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA.

Tkachenko, M.; and Lauw, H. W. 2016. Plackett-luce regression mixture model for heterogeneous rankings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 237–246.

Vershynin, R. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272.

Wang, Z.; Gu, Q.; Ning, Y.; and Liu, H. 2015. High dimensional em algorithm: Statistical optimization and asymptotic normality. *Advances in neural information processing systems*, 28.

Wu, C.; Yang, C.; Zhao, H.; and Zhu, J. 2016. On the convergence of the em algorithm: A data-adaptive analysis. *arXiv preprint arXiv:1611.00519*.

Wu, C. J. 1983. On the convergence properties of the EM algorithm. *The Annals of statistics*, 95–103.

Zhang, A. Y.; and Zhou, H. H. 2022. Leave-one-out Singular Subspace Perturbation Analysis for Spectral Clustering. *arXiv preprint arXiv:2205.14855*.

Zhao, Z.; Liu, A.; and Xia, L. 2020. Learning Mixtures of Plackett-Luce Models with Features from Top-l Orders. *arXiv preprint arXiv:2006.03869*.

Zhao, Z.; Piech, P.; and Xia, L. 2016. Learning mixtures of Plackett-Luce models. In *International Conference on Machine Learning*, 2906–2914. PMLR.

Zhao, Z.; Villamil, T.; and Xia, L. 2018. Learning mixtures of random utility models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zhao, Z.; and Xia, L. 2018. Composite marginal likelihood methods for random utility models. In *International Conference on Machine Learning*, 5922–5931. PMLR.

Zhao, Z.; and Xia, L. 2019. Learning mixtures of plackett-luce models from structured partial orders. *Advances in Neural Information Processing Systems*, 32.

Zitnik, M.; and Zupan, B. 2012. Nimfa: A Python Library for Nonnegative Matrix Factorization. *Journal of Machine Learning Research*, 13: 849–853.

Technical Appendix: Efficient and Accurate Learning of Mixtures of Plackett-Luce Models

Duc Nguyen, Anderson Y. Zhang

March 23, 2023

Contents

A Preliminaries

In this section, we introduce extra definitions and notations that will be used throughout the rest of the paper and some useful high-dimension concentration inequalities. Many of these results apply not only to the Plackett-Luce model but also other *random utility models* (see Definition 1). We believe these results can be of independent interest.

A.1 Useful Concentration Inequalities

As a shorthand notation, $a \lesssim b$ means that $a \leq Cb$ for some constant C and $a \asymp b$ means that $a \lesssim b, b \lesssim a$.

Definition 1 (Random utility model generative process)**.** *A random utility model of ranking data is parametrized by utilities $\theta^* = [\theta_1^*, \ldots, \theta_n^*]$ and noise distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$. A ranking π is drawn by first drawing $\epsilon_i \sim \mathcal{D}_i$ independently. Define $U_i = \theta_i^* + \epsilon_i$ for $i \in [n]$. $\pi = \arg\operatorname{sort} U$.*

When all the noise distributions are the same $\mathcal{D}_i = \mathcal{D}$ for some distribution $\mathcal{D}$, the ranking distribution is also referred to as a random utility model with independently and identically distributed noise, or IID-RUM. The PL model is an IID-RUM with a standard Gumbel noise distribution. The Thurstone model [Thurstone, 1927] is another IID-RUM but with $\mathcal{N}(0, \frac{1}{2})$ being the noise distribution.

Definition 2 (Sub-gaussian random variable, Proposition 2.5.2 of Vershynin [2018])**.** *A random variable $X \in \mathbb{R}$ is a sub-gaussian random variable with variance parameter σ if the tails of X satisfies*

$$\mathbb{P}\Big(|X - \mathbb{E}X| \geq t\Big) \leq 2\exp\left(-\frac{t^2}{\sigma^2}\right)$$

and the sub-gaussian norm of X is defined as

$$\|X\|_{\psi_2} = \inf\{t > 0 \,:\, \mathbb{E}\exp\left((X - \mathbb{E}X)^2/t^2\right) \leq 2\}.$$

Definition 3 (Sub-gaussian distribution in high dimension, Definition 3.4.1 of Vershynin [2018])**.** *A random vector $X \in \mathbb{R}^d$ is a sub-gaussian random variable if the one-dimensional marginals $X^\top x$ are sub-gaussian random variables for all $x \in \mathbb{R}^d$. The sub-gaussian norm of X is defined as*

$$\|X\|_{\psi_2} = \sup_{x : \|x\|_2 = 1} \|X^\top x\|_{\psi_2}\,.$$

If a random vector X is a sub-gaussian random vector with sub-gaussian norm σ, we write $X \sim \mathrm{SG}(\sigma^2)$.

Lemma 1 (Sub-gaussian norm of permutation vector, Proposition 3 of Shah and Song [2018])**.** *Consider the pairwise vector representation of permutations described in Equation (7). Under any random utility model, there exists a constant τ such that the sub-gaussian norm of the random variable X from such embedding satisfies*

$$\|X\|_{\psi} \leq \sqrt{\tau n}\,.$$

Lemma 2. *(Concentration of norm for sub-gaussian random variables, Theorem 3.1.1 of Vershynin [2018]) Let $X = (X_1, \ldots X_n) \in \mathbb{R}^n \sim \mathcal{N}(0, \sigma_0 I_n)$. Then there exists a constant c such that*

$$\mathbb{P}\Big(\big|\|X\|_2 - \sqrt{n}\big| \geq t\Big) \;\leq\; 2\exp\left(-\frac{ct^2}{\sigma_0^4}\right)\,.$$

Theorem 3. *(Hoeffding's inequality) Let $X_1, X_2, \ldots, X_m$ be independent random variables such that $X_i \in [a_i, b_i]$ for $i \in [m]$. Then*

$$\mathbb{P}\Big(\big|\sum_{i=1}^m X_i - \mathbb{E}[X_i]\big| > t\Big) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^m (a_i - b_i)^2}\right)\,.$$

Theorem 4. *(Concentration inequality for a sum of Normal random variables) Let $X_1, \ldots, X_n$ be independent normal random variables distributed as $\mathbb{N}(0, \sigma^2)$. Then*

$$\mathbb{P}\left(|\sum_{i=1}^{n} X_i| > t \right) \leq 2\exp\left(-\frac{t^2}{2\sigma^2 n} \right) .$$

A.2 Mixtures of Sub-gaussian Distributions

We first introduce notations and theoretical quantities that bridge the gap between the analysis of mixtures of PL models and that of mixtures of sub-gaussian distributions. Let the means of K sub-gaussian distributions be $\mu_1^*, \ldots, \mu_K^*$ ($\mu_k^* \in \mathbb{R}^{\binom{n}{2}}$ in our algorithm). The observed pairwise vector X_l for $l \in [m]$ can be written as

$$X_l = \mu_{z_l}^* + \epsilon_l$$

where $\epsilon_l \in \mathbb{R}^{\binom{n}{2}}$ is a random noise vector with gaussian norm $\lesssim \sqrt{n}$ as shown earlier. Let $E = [\epsilon_1, \ldots, \epsilon_m] \in \mathbb{R}^{m \times \binom{n}{2}}$ be the concatenation of all the noise vectors. Let $P \in \mathbb{R}^{m \times \binom{n}{2}}$ be the concatenation of the 'centers' of the m random vectors: $P = [\mathbb{E}[X_1], \ldots, \mathbb{E}[X_m]]^\top$. In short,

$$X = P + E$$

Let $\Delta = \min_{k \neq k'} \|\mu_k^* - \mu_{k'}^*\|_2$ denote the minimum inter-center gap.

The following lemma will be useful in deriving a bound on the operator norm of E. This lemma is often found in the analysis of random graphs such as in stochastic block model.

Lemma 5. *[Theorem 5.2 of Lei and Rinaldo [2015]] Let A be the adjacency matrix of a random graph on n^* node in which edges occur independently. Set $P = \mathbb{E}[A] = (p)_{ij}$ and assume that $\max_{ij} p_{ij} \leq \frac{d}{n^*}$ for some $d \geq c_0 \log n$ and $c_0 > 0$. Then for any $r > 0$, there is a constant C that depends only on r, c_0 such that*

$$\|E\|_{op} = \|A - P\|_{op} \leq C\sqrt{d}$$

with probability at least $1 - n^{*-r}$.

Lemma 6. *Under the generative model for a mixture of K random utility models, there exists a constant C_E such that the operator norm of the error matrix satisfies*

$$\|E\|_{op} \leq C_E \sqrt{n}(\sqrt{m+n})$$

with probability at least $1 - (m+n)^{-10}$.

Proof. Note that in the broad class of random utility models, the distribution of the comparisons of two *disjoint* pairs are independent. Building on this idea, we partition $E^\top$ into $O(n)$ blocks of size $O(n) \times m$ where each block consists of entries corresponding to pairs that are mutually disjoint. This is so that within

each block, the entries (comparisons) are independent. Specifically, consider the set of all pairs

$$\{(1,2),(1,3),\ldots,(1,n),(2,3),\ldots,(2,n),\ldots,(n,n-1)\}\,.$$

Imagine that we place all these pairs into the upper diagonal entries of a $n \times n$ matrix as shown in Figure 1.

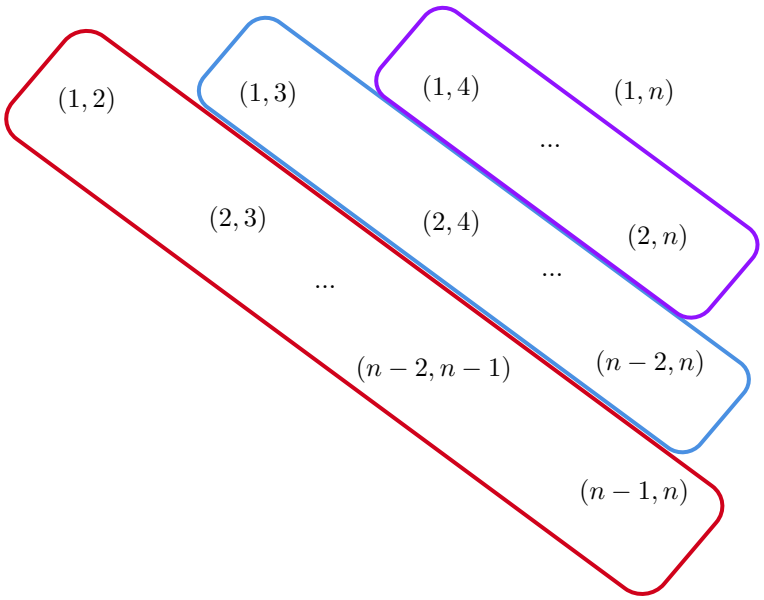


Figure 1: $\binom{n}{2}$ pairs are arranged into the upper diagonal entries of a square matrix with the super-diagonals highlighted.

Now for each super-diagonal of this matrix, partition the entries into half by taking every other item. For example, for the first super-diagonal, partition the entries as follows (WLOG assuming that n is even) to obtain two blocks of entries.

$$\{(1,2),(3,4),\ldots,(n-1,n)\} \cup \{(2,3),(4,5),\ldots,(n-2,n-1)\}\,.$$

Repeat the same procedure for all super-diagonals and we obtain $2(n-1)-1$ of these sets. Note that these subsets are partitions of the pairs (columns of E), we can rewrite E as a concatenation of block matrices, each corresponding to a

subset of pairs.

$$E^\top = \begin{bmatrix} E_1^\top \\ E_2^\top \\ \vdots \\ E_{O(n)}^\top \end{bmatrix}.$$

Each submatrix E_i is of dimension $O(n) \times m$. Note that permutation of the columns of E does not change the operator norm. It follows that:

$$\|E\|_{op} = \|E^\top\|_{op} \leq \sqrt{2n} \max_i \|E_i\|_{op}.$$

To bound $\|E_i\|_{op}$, we need to use Lemma 5 with $d = n$. Using the symmetric dilation trick,

$$\|E_i\|_{op} = \|\mathcal{S}(E_i)\|_{op} = \left\| \begin{bmatrix} 0 & E_i \\ E_i^\top & 0 \end{bmatrix} \right\|_{op}.$$

The matrix $\mathcal{S}(E_i)$ is a symmetric, binary matrix of size $O((n+m)) \times O((n+m))$ with independently distributed entries (like an adjacency matrix). By Lemma 5, there exists a constant C_E such that, with probability at least $1 - (n+m)^{-10}$,

$$\|E_i\|_{op} = \|\mathcal{S}(E_i)\|_{op} \leq 1/\sqrt{2}\, C_E \sqrt{n+m}.$$

This completes the proof. ■

The following theorem provides guarantees on the performance of the spectral clustering algorithm with adaptive dimension reduction (Algorithm 1) when applied to a mixture of K random utility models. Before showing the theorem, let us introduce some definitions. Let $\xi = \frac{1}{m/K} \min_{k \in [K]} |\{l : z_l^* = k\}$ and $\xi m/K$ is the smallest cluster size. Assuming a mixture model with uniform mixing distributions, when m is sufficiently large, $\xi \approx 1$. Note that the clusters are identifiable up to a change in label. Let Φ denote the set of all re-labelling function. The mis-clustering rate of an assignment z is defined as

$$\ell(z, z^*) = \min_{a \in \Phi} \frac{1}{m} \sum_{l=1}^{m} \mathbb{1}[a(z_l) \neq z_l^*].$$

Theorem 7 (Theorem 3.2 of Zhang and Zhou [2022])**.** *Consider the generative model described above. Assume that $\epsilon_l \sim SG(\sigma^2)$ independently with mean zero for $l \in [m]$. Asssume that $\beta m/K^4 \geq 400$. There exist constants C, C', C_1, C_2 such that under the assumption that*

$$\psi := \frac{\sqrt{\xi}\sqrt{m}\Delta}{K^2 \|E\|_{op}} > C,$$

and $\rho = \frac{T}{\|E\|_{op}}$ *satisfies* $C_1 \le \rho \le \frac{\psi}{C_2}$*, then the output* z *of Algorithm 1 satisfies*

$$\mathbb{E}\ell(z, z^*) \le \exp\left(-(1 - C'(\rho\psi^{-1} + \rho^{-1}))\frac{\Delta^2}{8\sigma^2}\right) + \exp\left(-\frac{m}{2}\right) .$$

If we further assume that $\psi, \rho \to \infty$ *and* $\rho/\psi = o(1)$*, then*

$$\mathbb{E}[\ell(\hat{z}, z^*)] \le \exp\left(-(1 - o(1))\frac{\Delta^2}{8\sigma^2}\right) + \exp\left(-m/2\right) .$$

B Spectral Initialization

This section is dedicated to proving Theorem 1. At a high level, the proof consists of three main components. Firstly, we want to show that under the generative model described in Theorem 1, the inter-center gap is large. This allows us to show that the mis-clustering error incurred by Algorithm 1 is small using Theorem 7. The second component is the finite sample error bound for the least squares estimator in Algorithm 2. Lastly, we combine the two results to show that under the generative model in the theorem statement, the pairwise probability estimates obtained from the clusters produced by the spectral clustering algorithm are accurate and therefore Algorithm 2 and, by extension, Algorithm 3 also return accurate parameter estimates.

B.1 The Inter-center Gap

Define κ to be the dynamic range, i.e., $\kappa := \max_{i\in[n],k\in[K]} |\theta_i^{*k}|$. In our generative model, $\kappa \lesssim \sqrt{\log n}$. The dynamic range is the reason why the term $\exp\left(O(\sqrt{\log n})\right)$ appears in our bound. The rest of this section is devoted to showing that the inter-center gap Δ is sufficiently large under said generative model. The following theorem shows that in the special case when $L = n$, the mixture model in Theorem 1 has a large inter-center gap (with high probability).

Theorem 1. *Fix a constant number of mixture components K. Suppose that for mixture component $k \in [K]$, $\theta^{*k} \sim \mathbb{N}(0, I_n)$. For a sufficiently large n, there exist constants $C, c > 0$ such that*

$$\Delta \geq \frac{c}{\exp\left(C\sqrt{\log n}\right)} \cdot \sqrt{n^2 - n\log n}$$

with probability at least $1 - O(\frac{K^2}{n^{10}}) - \exp\left(O(-n)\right)$.

Proof. We restrict our analysis to just two mixture components θ and θ' for now. To obtain the high probability guarantee for K mixture components, a simple union bound argument suffices.

The mapping function under the case Plackett-Luce is exactly the link function for the BTL model. Specifically, for items i, j with partworths θ_i^*, θ_j^* respectively,

$$P_{ij} = \frac{1}{1 + \exp\left(-(\theta_i^* - \theta_j^*)\right)}.$$

The separation of the two pairwise centers satisfies

$$\begin{aligned}\Delta^2 &= \sum_{i<j} (P_{ij} - P'_{ij})^2 \\ &\geq \sum_{i<j} l(\kappa)^2 \cdot \left((\theta_i^* - \theta^*) - (\theta_i^{*\prime} - \theta_j^{*\prime})\right)^2,\end{aligned}$$

where $l(\kappa)$ satisfies

$$\left|\frac{1}{1+\exp(-x)} - \frac{1}{1+\exp(-y)}\right| \geq l(\kappa)\cdot|x-y| \,\forall\, x,y \in [-2\kappa,+2\kappa]\,.$$

Clearly, $l(\kappa)$ is dependent on the dynamic range κ. Since the logit function is continuous everywhere, we can lower bound $l(\kappa)$ by lower bounding the gradient of the logit function within the dynamic range, which is

$$\frac{\exp(-x)}{(1+\exp(-x))^2}$$

It is well known that for $\epsilon_1,\ldots,\epsilon_n \sim N(0,1)$, there exists a constant C such that $\max_i|\epsilon_i| \leq 2C\sqrt{\log n}$ with probability at least $1-\frac{1}{n^{10}}$. There also exists a constant $c'>0$ such that

$$l(\kappa) \geq \frac{\exp\left(-2C\sqrt{\log n}\right)}{(1+\exp\left(-2C\sqrt{\log n}\right))^2} \geq c'\exp\left(-2C\sqrt{\log n}\right)\,.$$

We will now tackle the remaining term in the expression for Δ^2. We have

$$\begin{aligned}
&\sum_{i<j}\left((\theta_i^*-\theta_j^*)-(\theta_i^{*\prime}-\theta_j^{*\prime})\right)^2\\
&=\frac{1}{2}\sum_{i\neq j}\left((\theta_i^*-\theta_j^*)-(\theta_i^{*\prime}-\theta_j^{*\prime})\right)^2\\
&=\frac{1}{2}\sum_{i\neq j}\left((\theta_i^*-\theta_i^{*\prime})-(\theta_j^*-\theta_j^{*\prime})\right)^2\\
&=\frac{1}{2}\sum_{i\neq j}(\theta_i^*-\theta_i^{*\prime})^2+(\theta_j^*-\theta_j^{*\prime})^2-2(\theta_i^*-\theta_i^{*\prime})(\theta_j^*-\theta_j^{*\prime})\\
&=(n-1)\cdot\|\theta^*-\theta^{*\prime}\|_2^2-(\theta^*-\theta^{*\prime})^\top(\mathbf{1}_n\mathbf{1}_n^\top-I_n)(\theta^*-\theta^{*\prime})\\
&=n\cdot\|\theta^*-\theta^{*\prime}\|_2^2-(\theta^*-\theta^{*\prime})^\top(\mathbf{1}_n\mathbf{1}_n^\top)(\theta^*-\theta^{*\prime})\\
&=n\cdot\|\theta^*-\theta^{*\prime}\|_2^2-\left((\theta^*-\theta^{*\prime})^\top\mathbf{1}_n\right)^2 \quad (*)\,.
\end{aligned}$$

For the second term, note that $\theta^*-\theta^{*\prime}\sim\mathcal{N}(0,2I_n)$. Invoking Theorem 4, there exists a constant c'' (e.g., 80) such that

$$(\theta^*-\theta^{*\prime})^\top\mathbf{1}_n \leq \sqrt{c''\,n\log n}$$

with probability n^{-10}.

As for the $\|\theta^*-\theta^{*\prime}\|_2$ term. Applying Lemma 2 gives

$$\|\theta^*-\theta^{*\prime}\|_2^2 \geq 0.9n$$

with probability $1-O(\exp(-n))$.

Going back to the display $(*)$, we have

$$\Delta \geq \exp\left(-C\sqrt{\log n}\right) \cdot \sqrt{0.9n^2 - c''\, n \log n}$$

with probability at least $1 - O(n^{-10}) - O(\exp(-n))$. Applying union bound over all pairs of mixture components in $[K]$ gives us the conclusion in the theorem statement. ■

The following theorem generalizes the previous theorem to a general L.

Theorem 2. *Fix a constant number of mixture components K. Suppose that for mixture component $k \in [K]$, $\theta^{*k}_{1:L} \sim \mathbb{N}(0, I_L)$ and $\theta^{*k}_{L+1:n} = 0$. There exist constants $C, c > 0$ such that*

$$\Delta \geq \frac{c}{\exp\left(C\sqrt{\log n}\right)} \cdot \sqrt{nL - L\log n}$$

with probability at least $1 - O(\frac{K^2}{n^{10}}) - O(K^2 \exp(-L))$.

Proof. Following the same argument as in the proof of Theorem 1, there exists a constant $c' > 0$ such that we decompose the gap between two cluster centers as

$$\begin{aligned}
\Delta^2 &= \sum_{i<j} (P_{ij} - P'_{ij})^2 \\
&\geq \sum_{i<j} l(\kappa)^2 \cdot \left((\theta^*_i - \theta^*_j) - (\theta^{*\prime}_i - \theta^{*\prime}_j)\right)^2, \\
&= \left(\exp\left(-C\sqrt{\log n}\right)\right)^2 \cdot \left(n\left\|\theta^* - \theta^{*\prime}\right\|_2^2 - \left((\theta^* - \theta^{*\prime})^\top \mathbf{1}_n\right)^2\right).
\end{aligned}$$

Note that by design,

$$\theta^* - \theta^{*\prime} = \left[(\theta^* - \theta^{*\prime})_{1:L}, 0, \ldots, 0\right].$$

Applying Lemma 2, we have

$$\|\theta^* - \theta^{*\prime}\|_2 = \|(\theta^* - \theta^{*\prime})_{1:L}\|_2 \geq \sqrt{0.9L}$$

with probability at least $1 - O(\exp(-L))$.

For the second term, by Theorem 4, there exists a constant c such that

$$(\theta^* - \theta^{*\prime})^\top \mathbf{1}_n = (\theta^* - \theta^{*\prime})^\top_{1:L} \mathbf{1}_L < \sqrt{c\, L \log n}$$

with probability at least $O(n^{-10})$. The rest of the proof proceeds similarly to that of Theorem 1. ■

B.2 The Least Squares Estimator

In this section, we obtain a general finite sample error guarantees for the least squares estimator (Algorithm 2). Recall that Algorithm 2 apply a mapping function before solving a constrained least squares optimization problem. The mapping function used in the algorithm, the logit function, is specific to the Plackett-Luce/BTL model. If we change the mapping function to, say the inverse Normal CDF, we can apply Algorithm 2 to parameter estimation under the Thurstone model [Thurstone, 1927] (See Definition 1). We believe that these results can be extended to other ranking or paired comparison models, or to an incomplete observation setting where certain pairs of items are not compared.

Lemma 3. *Let $\hat{\phi} \in \mathbb{R}^{n \times n}$ be the transformed pairwise measurement in Algorithm 2 and ϕ^* be its ideal value. Then the output θ of the algorithm satisfies*

$$\|N(\theta) - N(\theta^*)\|_2^2 \leq \frac{\|\hat{\phi} - \phi^*\|_F^2}{2n}\,.$$

Proof. We first rewrite the least squares optimization problem in a more convenient form. Let $\mathrm{vec}(\phi^*)$ denote the vectorization of ϕ^* (row major order). There exists a matrix $Z \in \{0, 1, -1\}^{n(n-1) \times n}$ such that $\mathrm{vec}(\phi^*) = Z\theta^*$ for any θ^*. Specifically, $Z = [Z_1, \ldots, Z_n]$ is a block matrix consisting of of n blocks of dimension $n - 1 \times n$ as shown below.

$$\begin{bmatrix} \phi_{12}^* \\ \phi_{13}^* \\ \cdots \\ \phi_{1n}^* \\ \phi_{21}^* \\ \cdots \\ \phi_{n,n-1}^* \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ \cdots \\ Z_n \end{bmatrix} \theta^*\,,$$

where

$$Z_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & -1 & 0 & \ldots & 0 \\ 1 & 0 & 0 & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & 0 & 0 & \ldots & -1 \end{bmatrix} \in \mathbb{R}^{n-1 \times n}\,,$$

$$Z_2 = \begin{bmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 & \dots & -1 \end{bmatrix} \in \mathbb{R}^{n-1 \times n},$$

and so on. One can see that the Z_i's are different column-permuted versions of Z_1. Namely, there exist permutation matrices $R_1, \dots, R_{n-1}$ such that $Z_2 = Z_1 R_1, Z_3 = Z_1 R_1 R_2$, .etc and $Z_n = Z_1 R_1 \dots R_{n-1}$. We can also check that $Z_1 \mathbf{1}_n = 0$.

Under our new notations, the least squares optimization problem in Algorithm 2 is

$$\hat{\theta} = \arg\min_{\theta} \|\mathrm{vec}(\hat{\phi}) - Z\theta\|_2 .$$

It is well known (cf., Strang et al. [1993]) that the least squares solution is

$$\hat{\theta} = (Z^\top Z)^\dagger Z^\top \mathrm{vec}(\hat{\phi}) ,$$

where $(Z^\top Z)^\dagger$ is the pseudo-inverse of $Z^\top Z$. On the other hand, we also have

$$\theta^* = (Z^\top Z)^\dagger Z^\top \mathrm{vec}(\phi^*) .$$

Combining the above two equalities gives

$$\hat{\theta} - \theta^* = (Z^\top Z)^\dagger Z^\top (\mathrm{vec}(\hat{\phi}) - \mathrm{vec}(\phi^*)) .$$

Now, since we care about the normalized ℓ_2 norm between θ and θ^*, we have

$$\begin{aligned}
\|N(\hat{\theta}) - N(\theta^*)\|_2^2 &= \|\hat{\theta} - \frac{1}{n}\mathbf{1}^\top \hat{\theta} \cdot \mathbf{1} - \theta^* + \frac{1}{n}\mathbf{1}^\top \theta^* \cdot \mathbf{1}\|_2^2 \\
&= \|\hat{\theta} - \theta^* + (\frac{1}{n}\mathbf{1}^\top \theta^* - \frac{1}{n}\mathbf{1}^\top \hat{\theta}) \cdot \mathbf{1}\|_2^2 \\
&= \|\hat{\theta} - \theta^*\|_2^2 - 2\langle \frac{1}{n}\mathbf{1}^\top (\hat{\theta} - \theta^*) \cdot \mathbf{1}, \hat{\theta} - \theta^* \rangle + \langle \frac{1}{n}\mathbf{1}^\top (\hat{\theta} - \theta^*) \cdot \mathbf{1}, \frac{1}{n}\mathbf{1}^\top (\hat{\theta} - \theta^*) \cdot \mathbf{1} \rangle \\
&= \|\hat{\theta} - \theta^*\|_2^2 - \frac{2}{n} \cdot ((\hat{\theta} - \theta^*)^\top \mathbf{1})^2 + \frac{1}{n^2} \cdot \mathbf{1}^\top \mathbf{1} ((\hat{\theta} - \theta^*)^\top \mathbf{1})^2 \\
&= \|\hat{\theta} - \theta^*\|_2^2 - \frac{1}{n} \cdot (\hat{\theta} - \theta^*)^\top \mathbf{1}\mathbf{1}^\top (\hat{\theta} - \theta^*) \\
&= (\theta - \theta^*)^\top (I - \frac{\mathbf{1}}{\sqrt{n}} \frac{\mathbf{1}}{\sqrt{n}})(\theta - \theta^*) \\
&= (\mathrm{vec}(\hat{\phi}) - \mathrm{vec}(\phi^*)) Z (Z^\top Z)^\dagger (I - \frac{\mathbf{1}}{\sqrt{n}} \frac{\mathbf{1}}{\sqrt{n}})(Z^\top Z)^\dagger Z^\top (\mathrm{vec}(\hat{\phi}) - \mathrm{vec}(\phi^*)) .
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|N(\hat{\theta}) - N(\theta^*)\|_2^2 &\leq \lambda_{\max}\left(Z(Z^\top Z)^\dagger (I - \frac{\mathbf{1}}{\sqrt{n}}\frac{\mathbf{1}^\top}{\sqrt{n}}) \underbrace{(Z^\top Z)^\dagger Z^\top}_{A^\top} \right) \cdot \|\hat{\phi} - \phi^*\|_F^2 \\
&= \lambda_{\max}\left(A(I - \frac{\mathbf{1}}{\sqrt{n}}\frac{\mathbf{1}^\top}{\sqrt{n}})A^\top \right) \cdot \|\hat{\phi} - \phi^*\|_F^2 \\
&= \lambda_{\max,\perp\!\!\!\perp}\left(A^\top A \right) \cdot \|\hat{\phi} - \phi^*\|_F^2 \\
&= \lambda_{\max,\perp\!\!\!\perp}\left((Z^\top Z)^\dagger Z^\top Z (Z^\top Z)^\dagger \right) \cdot \|\hat{\phi} - \phi^*\|_F^2 \\
&= \lambda_{\max,\perp\!\!\!\perp}((Z^\top Z)^\dagger) \cdot \|\hat{\phi} - \phi^*\|_F^2 \,,
\end{aligned}$$

where $\lambda_{\max,\perp\!\!\!\perp}$ is the maximum eigenvalue associated with the eigenspace orthogonal to the vector $\mathbf{1}_n$. One can verify the following.

$$\begin{aligned}
Z^\top Z &= Z_1^\top Z_1 + Z_2^\top Z_2 + \ldots + Z_n^\top Z_n \\
&= \begin{bmatrix} n-1 & -1 & -1 & \ldots & -1 \\ -1 & 1 & 0 & \ldots & 0 \\ -1 & 0 & 1 & \ldots & 0 \\ \vdots & & & & \\ -1 & 0 & 0 & \ldots & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ -1 & n-1 & -1 & \ldots & -1 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & & & & \\ 0 & -1 & 0 & \ldots & 1 \end{bmatrix} + \cdots \begin{bmatrix} 1 & 0 & 0 & \ldots & -1 \\ 0 & 1 & 0 & \ldots & -1 \\ 0 & 0 & 1 & \ldots & -1 \\ \vdots & & & & \\ -1 & -1 & -1 & \ldots & n-1 \end{bmatrix} \\
&= 2n\, I_n - 2n\, \frac{\mathbf{1}}{\sqrt{n}}\frac{\mathbf{1}^\top}{\sqrt{n}} \,.
\end{aligned}$$

Hence, $\lambda_{\min,\perp\!\!\!\perp}(Z^\top Z) = 2n$ and $\lambda_{\max,\perp\!\!\!\perp}((Z^\top Z)^\dagger) = \frac{1}{2n}$. This directly yields the inequality in the theorem statement. ■

B.3 Proof of Theorem 1

Let us introduce some notations. Let $C^{*k} = \{l \in [m] \,|\, z_l^* = k\}$ for $k \in [K]$ denote the true cluster assignment. To avoid cluttering the notation, let us suppress the relabeling operator here and consider $\hat{C}^k = \{l \in [m] \,|\, z_l = k\}$ to be the cluster assignment produced by Algorithm 1 corresponding to mixture component k.

As noted before, when the mixing probabilities are uniform, $\beta^* = \frac{1}{K}\mathbf{1}$, $\xi \approx 1$ and the size of each true cluster is $\approx \frac{m}{K}$ whp. In fact, for a cluster k, Hoeffding's inequality gives

$$\mathbb{P}\left(\left| |C^{*k}| - \frac{m}{K} \right| > t \right) \leq 2\exp\left(-\frac{2t^2}{m} \right) .$$

Hence, $\frac{0.9m}{K} \leq |C^{*k}| \leq \frac{1.1m}{K}$ with probability at least $1 - O(\exp\left(-m/K^2\right))$. Applying union bound over all clusters, we conclude that the event

$$\mathcal{E}_0 = \left\{ |C^{*k}| \in \left[\frac{0.9m}{K}, \frac{1.1m}{K} \right] \forall k \in [K] \right\}$$

happens with probability at least $1 - O(K\exp\left(-m/K^2\right))$.

On the choice of T. Theorem 6 gives $\|E\|_{op} \lesssim \sqrt{n}(\sqrt{m} + \sqrt{n})$ with probability at least $1 - \frac{1}{(n+m)^{10}}$. The technical condition in Zhang and Zhou [2022] sugggests that T is at least a constant factor larger than $\|E\|$. It therefore suffices to pick $T = \sqrt{n}\sqrt{m+n}\sqrt{\log n}$. Additionally, the second bound in Theorem 6 holds if the following event holds

$$\mathcal{E}_0' = \left\{ \frac{\sqrt{\log n}\, K^2 \|E\|_{op}}{\xi\sqrt{m}\Delta} = o(1) \right\}.$$

We will show that this is satisfied under the generative model described in Theorem 1. For now, let us define the following events.

$$\mathcal{E}_1 = \left\{ \Delta \geq \frac{\sqrt{d}}{\exp\left(D/2\sqrt{\log n}\right)} \sqrt{L}\sqrt{n - \log n} \right\} \cup \mathcal{E}_0 \cup \mathcal{E}_0'.$$

$$\mathcal{E}_2 = \left\{ \Delta \geq \frac{\sqrt{d}}{\exp\left(D/2\sqrt{\log n}\right)} \sqrt{n^\alpha}\sqrt{n - \log n} \right\} \cup \mathcal{E}_0 \cup \mathcal{E}_0'.$$

d and D are the constants in Theorem 2. Recall the sub-gaussian norm of the pairwise representation of permutations in Lemma 1. For the rest of the proof, we use the same τ.

Lemma 4. *Suppose that event $\mathcal{E}_1$ or event $\mathcal{E}_2$ holds and $m \geq cK^4$ or a sufficiently large constant c. Then the clustering $\hat{z}$ output of Algorithm 1 satisfies the following.*

$$\begin{aligned}
&\mathbb{P}\left(m \cdot \ell(\hat{z}, z^*) > \frac{m}{K} \exp\left(-\frac{\Delta^2}{16\tau n} \right) \,\middle|\, \mathcal{E}_1 \right) \\
&\leq K \cdot \exp\left(-(1 - o(1)) \frac{\Delta^2}{16\tau n} \right) + K \cdot \exp\left(-\frac{m}{2} + \frac{\Delta^2}{16\tau n} \right),
\end{aligned}$$

Proof. Under the assumptions stated in the lemma, the output of Algorithm 1 achieves exponentially decaying mis-clustering error rate and

$$\begin{aligned}
&\mathbb{P}\left(m \cdot \ell(z, z^*) > t \,\middle|\, \mathcal{E}_1 \right) = \mathbb{P}\left(\ell(z, z^*) > \frac{t}{m} \right) \\
&\leq \frac{m}{t} \cdot \exp\left(-(1 - o(1)) \frac{\Delta^2}{8\sigma^2} \right) + \frac{m}{t} \cdot \exp\left(-\frac{m}{2} \right),
\end{aligned}$$

where the last step comes from applying Markov's inequality and ψ, ρ, C' are defined in Theorem 7 and $\sigma^2 = \tau n$. The theorem statement is obtained by replacing t with $\frac{m}{K} \cdot \exp\left(-\frac{\Delta^2}{16\tau n} \right)$. ■

Theorem 1. *Consider a mixture of K Plackett-Luce models with uniform mixing probabilities. Suppose that $\theta_i^{*k} = 0 \, \forall i \in [L+1:n]$ and $\theta_{1:L}^{*k} \sim \mathbb{N}(0, I_L)$ for $k \in [K]$. Fix a constant $\alpha > 0$. There exist constants c, c_1, C_1, C_2 such that if $m \geq c \max\{K^4, Kn\}$, the output $\hat{\boldsymbol{\theta}}$ of Algorithm 3 satisfies the following. If $L \geq c_1 \exp\left(C_1 \sqrt{\log n}\right)$,*

$$dist(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^*) = O\left(\exp\left(O(\sqrt{\log n})\right) \cdot \left(\sqrt{\frac{K^2 n \log n}{m}} + \frac{\sqrt{Kn}}{e^{L^{0.99}}}\right)\right)$$

with probability $1 - O(\frac{K}{n^8}) + O(K^2 n^2 \exp\left(-L^{0.99}\right))$. If $L \geq C_2 n^\alpha$ and assuming that $n = \omega(\log m)$,

$$dist(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^*) = O\left(\exp\left(O(\sqrt{\log n})\right) \sqrt{\frac{K^2 n \log n}{m}}\right)$$

with probability $1 - O(\frac{K}{n^8}) + O(K^2 n^2 \exp\left(-n^\alpha\right))$.

Proof. Since the output of Algorithm 1 are K clusters which are then used to estimate the pairwise preference probabilities – the input to Algorithm 2 – it suffices to show that each pairwise center $\hat{P}^k$ is close to the corresponding true centers P^{*k}.

Focusing on a cluster $k \in [K]$ and a single pair (i, j), let N_{ij}^k denote the number of times that item j 'beats' item i among the permutations in cluster $\hat{C}^k$.

$$N_{ij}^k = \sum_{l \in C^{*k}} \mathbb{1}[\pi_l(i) < \pi_l(j)] - \sum_{l \in C^{*k}, l \notin \hat{C}^k} \mathbb{1}[\pi_l(i) < \pi_l(j)] + \sum_{l \notin C^{*k}, l \in \hat{C}^k} \mathbb{1}[\pi_l(i) < \pi_l(j)] \, .$$

We will analyze these terms separately. By Hoeffding's inequality, the first term can be bounded as

$$\mathbb{P}\left(\left| \sum_{l \in C^{*k}} \mathbb{1}[\pi_l(i) < \pi_l(j)] - |C^{*k}| \cdot P_{ij}^* \right| > t\right) \leq 2\exp\left(-\frac{2t^2}{|C^{*k}|}\right) \leq 2\exp\left(-\frac{2K\, t^2}{1.1m}\right) .$$

Therefore, there exists a constant d_1 such that

$$\left| \sum_{l \in C^{*k}} \mathbb{1}[\pi_l(i) < \pi_l(j)] - |C^{*k}| \cdot P_{ij}^* \right| \leq d_1 \sqrt{m/K \, \log n}$$

with probability at least $1 - n^{-10}$.

For the other two terms in N_{ij}^k, note that by definition,

$$\left| \sum_{l \in C^{*k}, l \notin \hat{C}^k} \mathbb{1}[\pi_l(i) < \pi_l(j)] \right| + \left| \sum_{l \notin C^{*k}, l \in \hat{C}^k} \mathbb{1}[\pi_l(i) < \pi_l(j)] \right| \leq m \cdot \ell(z, z^*) \, .$$

At this point, we consider the two regimes of L separately as the gap Δ differs substantially between the two regimes, leading to different bounds on the quantity above.

For the regime $L \geq c_1 \exp\left(C_1\sqrt{\log n}\right)$. By Theorem 2, the gap Δ satisfies the following, for some constants d, D.

$$\Delta^2 \geq \frac{d}{\exp\left(D\sqrt{\log n}\right)} L(n - \log n)$$

with probability at least $1 - \frac{K^2}{n^{10}} - O(K^2 \exp(-L))$. Because $\exp\left(\sqrt{\log n}\right) = \omega(\log n)$, the ratio ρ/ψ in Theorem 7 satisfies

$$\frac{\sqrt{\log n}\, K^2 \|E\|_{op}}{\xi\sqrt{m}\Delta} \asymp \frac{\sqrt{\log n}\, K^2 \sqrt{n}\sqrt{m+n}}{\exp\left((C_1 - D)\sqrt{\log n}\right)\, \sqrt{n - \log n}\sqrt{m}} = o(1)\,.$$

Then by Theorem 2, Lemma 6 and considering the generative model in the theorem statement, event $\mathcal{E}_1$ happens with probability at least $1 - \frac{K^2}{n^{10}} - O(K^2 \exp(-L)) - O(K \exp\left(-m/K^2\right)) - \frac{1}{(n+m)^{10}}$. The last two terms in the probability guarantee diminish much faster than the other terms so we can simplify as

$$\mathbb{P}(\mathcal{E}_1) \geq 1 - O(\frac{K^2}{n^{10}}) - O(K^2 \exp(-L))\,.$$

By Lemma 4,

$$\begin{aligned}
&\mathbb{P}\left(m \cdot \ell(\hat{z}, z^*) > \frac{m}{K} \exp\left(-\frac{\Delta^2}{16\tau n}\right) \,\middle|\, \mathcal{E}_1\right) \\
&\leq K \cdot \exp\left(-(1 - o(1))\frac{\Delta^2}{16\tau n}\right) + K \cdot \exp\left(-\frac{m}{2} + \frac{\Delta^2}{16\tau n}\right) \\
&= K \cdot \exp\left(-\frac{(1 - o(1))d}{16\tau} \cdot \frac{L(n - \log n)}{\exp\left(D\sqrt{\log n}\right)\, n}\right) + K \cdot \exp\left(-\frac{m}{2} + \frac{d}{16\tau} \cdot \frac{L(n - \log n)}{\exp\left(D\sqrt{\log n}\right)\, n}\right) \\
&= K \cdot \exp\left(-\frac{(1 - o(1))d}{16\tau} \cdot \frac{L}{\exp\left(D\sqrt{\log n}\right)}\right) + K \cdot \exp\left(-\frac{m}{2} + \frac{d(1 - o(1))}{16\tau} \cdot \frac{L}{\exp\left(D\sqrt{\log n}\right)}\right)
\end{aligned}$$

There exists a constant c such that if $m \geq cn$, then the second exponential term is bounded as

$$K \cdot \exp\left(-\frac{m}{2} + \frac{d(1 - o(1))}{16\tau} \cdot \frac{L}{\exp\left(D\sqrt{\log n}\right)}\right) \leq K \exp(-n)\,,$$

which is dominated by the first exponential term. Hence, we can simplify the

probability bound on the mis-clustering error as

$$\begin{aligned}
&\mathbb{P}\left(m \cdot \ell(\hat{z}, z^*) > \frac{m}{K} \exp\left(-\frac{\Delta^2}{16\tau n}\right) \,\middle|\, \mathcal{E}_1\right) \\
&\leq 2K \cdot \exp\left(-\frac{(1-o(1))d}{16\tau} \cdot \frac{L}{\exp\left(D\sqrt{\log n}\right)}\right) \\
&= 2K \cdot \exp\left(-\frac{(1-o(1))dc_1}{16\tau} \cdot \exp\left((C_1 - D)\sqrt{\log n}\right)\right) \\
&\text{[There exist values for constants } c_1', C_1' \text{ such that if} \\
&c_1 > c_1', C_1 > C_1' \text{ then the above term is bounded by]} \\
&\leq 2K \cdot \exp\left(-c_1^{0.99} \exp\left(0.99 C_1 \sqrt{\log n}\right)\right) \\
&\leq 2K \cdot \exp\left(-L^{0.99}\right)
\end{aligned}$$

Conditioned on event $\mathcal{E}_1$ and with the choice of c_1, C_1 above, the following happens with probability at least $1 - 2K \cdot \exp\left(-L^{0.99}\right) - \frac{1}{n^{10}}$ for a single pair i, j and cluster k.

$$\left|N_{ij} - \frac{m}{K} \cdot P_{ij}^{*k}\right| \lesssim \sqrt{m/K \, \log n} + m\ell(z, z^*) \asymp \sqrt{m/K \, \log n} + \frac{m}{K e^{L^{0.99}}} \, .$$

Since $C^{*k} \asymp \frac{m}{K}$ and $|\hat{C}^k - C^{*k}| \leq \frac{m}{K e^{L^{0.99}}}$, we have $\hat{C}^k \asymp \frac{m}{K}$. Consequently,

$$|\hat{P}_{ij}^k - P_{ij}^{*k}| = \left|\frac{N_{ij}^k}{|\hat{C}^k|} - P_{ij}^*\right| \asymp \frac{1}{m/K} \cdot \left|N_{ij}^k - m/K \cdot P_{ij}^*\right| \lesssim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} + \frac{1}{e^{L^{0.99}}} \, .$$

We apply union bound over all pairs $i \neq j$. Conditioned on event $\mathcal{E}_1$, with probability at least $1 - \frac{2Kn^2}{\exp(L^{0.99})} - \frac{1}{n^8}$.

$$|\hat{P}_{ij}^k - P_{ij}^{*k}| \lesssim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} + \frac{1}{e^{L^{0.99}}}$$

for all pairs i, j in cluster k. Apply a union bound argument over all clusters, the above bound holds for all $k \in [K]$ and all pairs with probability at least $1 - \frac{2K^2n^2}{\exp(L^{0.99})} - \frac{K}{n^8}$. Since the event $\mathcal{E}_1$ happens with probability at least $1 -$

$O(\frac{K^2}{n^{10}}) - O(K^2\exp(-L))$, we can show that

$$\begin{aligned}
&\mathbb{P}\bigg(|\hat{P}_{ij}^k - P_{ij}^{*k}| \gtrsim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} + \frac{1}{e^{L^{0.99}}} \quad \forall k \in [K], i \neq j \in [n]\bigg) \\
&\leq \mathbb{P}(\sim \mathcal{E}_1) + \mathbb{P}(\mathcal{E}_1) \cdot \mathbb{P}\bigg(|\hat{P}_{ij}^k - P_{ij}^{*k}| \gtrsim \frac{\sqrt{K}\sqrt{\log}}{\sqrt{m}} + \frac{1}{e^{L^{0.99}}} \quad \forall k \in [K], i \neq j \in [n] \,\bigg|\, \mathcal{E}_1\bigg) \\
&\leq O(\frac{K^2}{n^{10}}) + O(K^2\exp(-L)) \\
&+ (1 - O(\frac{K^2}{n^{10}}) - O(K^2\exp(-L))) \cdot \left(\frac{2K^2n^2}{\exp(L^{0.99})} + \frac{K}{n^8}\right) \\
&\leq O(\frac{K}{n^8}) + O(K^2n^2\exp\left(-L^{0.99}\right)).
\end{aligned}$$

For the regime $L \geq C_2 n^\alpha$. Again by Theorem 1,

$$\Delta^2 \geq \frac{d}{\exp\left(D\sqrt{\log n}\right)} \cdot (nL - L\log n) \asymp \frac{dC_2n^\alpha}{\exp\left(D\sqrt{\log n}\right)} \cdot C_2n^\alpha \cdot (n - \log n)$$

with probability at least $1 - \frac{K^2}{n^{10}} - O(K^2\exp(-C_2n^\alpha))$. One can check that $\rho/\psi = o(1)$ since $\exp\left(D\sqrt{\log n}\right) = o(n^\alpha)$ for any $\alpha > 0$. Therefore, event $\mathcal{E}_2$ happens with probability at least $1 - \frac{K^2}{n^{10}} - O(K^2\exp(-C_2n^\alpha)) - O(K\exp\left(-m/K^2\right)) - \frac{1}{(n+m)^{10}}$. Following a similar argument as in the previous regime of L, we have

$$\begin{aligned}
&\mathbb{P}\bigg(m \cdot \ell(\hat{z}, z^*) > \frac{m}{K}\exp\left(-\frac{\Delta^2}{16\tau n}\right) \,\bigg|\, \mathcal{E}_2\bigg) \\
&\leq 2K \cdot \exp\left(-(1 - o(1))\frac{\Delta^2}{16\tau n}\right) \\
&= 2K \cdot \exp\left(-\frac{(1 - o(1))d}{16\tau} \cdot \frac{L(n - \log n)}{\exp\left(D\sqrt{\log n}\right)\, n}\right) \\
&= 2K \cdot \exp\left(-\frac{(1 - o(1))d}{16\tau} \cdot \frac{L}{\exp\left(D\sqrt{\log n}\right)}\right) \\
&\text{[There exist constants } C_2' \text{ such that if } C_2 > C_2' \text{ then} \\
&\text{the above term can be bounded as]} \\
&\leq 2K\exp(-n^\alpha)
\end{aligned}$$

By the assumption that $n = \omega(\log m)$, there also exist constants C_2'', such that if $C_2 > C_2''$ then $\frac{\Delta^2}{16\tau n} \asymp \exp(n^\alpha) \geq \sqrt{m}$ and $\exp\left(-\frac{\Delta^2}{16\tau n}\right) \leq \frac{1}{\sqrt{m}}$. Set $C_2 = \max\{C_2', C_2''\}$.
Conditioned on event $\mathcal{E}_2$ and with the choice of C_2 above, the following happens with probability at least $1 - 2K\exp(-n^\alpha) - \frac{1}{n^{10}}$ for a single pair i, j and cluster k.

$$\left|N_{ij} - \frac{m}{K} \cdot P_{ij}^{*k}\right| \lesssim \sqrt{m/K\, \log n} + m\ell(z, z^*) \asymp \sqrt{m/K\, \log n} + \sqrt{m/K}\,.$$

Clearly the first term dominates. Since $C^{*k} \asymp \frac{m}{K}$ and $|\hat{C}^k - C^{*k}| \leq \sqrt{\frac{m}{K}}$, we have $\hat{C}^k \asymp \frac{m}{K}$. Consequently,

$$|\hat{P}_{ij}^k - P_{ij}^{*k}| = \left| \frac{N_{ij}^k}{|\hat{C}^k|} - P_{ij}^* \right| \asymp \frac{1}{m/K} \cdot \left| N_{ij}^k - m/K \cdot P_{ij}^* \right| \lesssim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} .$$

We apply union bound over all pairs $i \neq j$ and cluster $k \in [K]$. Conditioned on event $\mathcal{E}_2$, with probability at least $1 - \frac{2K^2n^2}{\exp(-n^\alpha)} - \frac{K}{n^8}$, the following holds for all pairs in all cluster components.

$$|\hat{P}_{ij}^k - P_{ij}^{*k}| \lesssim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} .$$

Since event $\mathcal{E}_2$ happens with probability at least $1 - O(\frac{K^2}{n^{10}}) - O(K^2 \exp(-C_2 n^\alpha))$, we have

$$\begin{aligned}
&\mathbb{P}\left(|\hat{P}_{ij}^k - P_{ij}^{*k}| \gtrsim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} \quad \forall k \in [K], i \neq j \in [n] \right) \\
&\leq \mathbb{P}(\sim \mathcal{E}_2) + \mathbb{P}(\mathcal{E}_2) \cdot \mathbb{P}\left(|\hat{P}_{ij}^k - P_{ij}^{*k}| \gtrsim \frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} \quad \forall k \in [K], i \neq j \in [n] \,\middle|\, \mathcal{E}_2 \right) \\
&\leq O(\frac{K^2}{n^{10}}) + O(K^2 \exp(-n^\alpha)) \\
&+ (1 - O(\frac{K^2}{n^{10}}) - O(K^2 \exp(-C_2 n^\alpha))) \cdot \left(\frac{2K^2n^2}{\exp(-n^\alpha)} + \frac{K}{n^8} \right) \\
&= O(\frac{K}{n^8}) + O(K^2 n^2 \exp(-n^\alpha)) .
\end{aligned}$$

Bounding the Error introduced by the Logit Transformation. From Lemma 3, the output $\hat{\theta}$ satisfies

$$\|\hat{\theta}^k - N(\theta^{*k})\|_2 \leq \frac{\|\hat{\phi}^k - \phi^{*k}\|_F}{\sqrt{2n}} .$$

We have already obtain bound on $|\hat{P}_{ij}^k - P_{ij}^{*k}|$. We still need to account for the error introduced by the logit mapping function. By the sub-gaussian property, there exists a constant B' such that $|\theta_i| \leq B'\sqrt{\log n}$ $\forall i \in [n]$ with probability at least $1 - \frac{1}{n^8}$. Then for $B = 2B'$, we have for all pairs (i, j),

$$P_{ij}^* \geq \exp\left(-2B\sqrt{\log n} \right) .$$

Recall that $\exp\left(\sqrt{\log n}\right) = \frac{\sqrt{\log n}}{\sqrt{n}}$ and $\exp\left(\sqrt{\log n}\right) = o\big(\exp\left(\exp\left(\sqrt{\log n}\right)\right)\big)$. Therefore, there exist constants c, c_1''', C_1''' such that if $m \geq cKn$ and $L \geq c_1''' \exp\left(C_1''' \sqrt{\log n}\right)$, the following holds for all pairs (i, j) and clusters $k \in [K]$.

$$|\hat{P}_{ij}^k - P_{ij}^{*k}| \lesssim \frac{\sqrt{K \log n}}{\sqrt{m}} + \frac{1}{e^{\sqrt{L}}} \leq \frac{1}{2} \exp\left(-2B\sqrt{\log n} \right)$$

with probability at least $1 - O(\frac{K}{n^8}) - O(\frac{K^2n^2}{\exp(L^{0.99})})$. In other words,

$$\hat{P}_{ij}^k \geq \frac{1}{2}\exp\left(-2B\sqrt{\log n}\right) \quad \forall i \neq j \in [n], k \in [K]\,.$$

Recall the logit mapping function $\phi(\hat{p}) = \ln\left(\frac{\hat{p}}{1-\hat{p}}\right)$. Its gradient is

$$\frac{1}{\hat{p}} + \frac{1}{1-\hat{p}}\,.$$

Since the logit function is continuous, an upper bound on its function also suffices as a Lipschitz constant. Namely, for any $p, p' \in [\frac{1}{2}\exp\left(-2B\sqrt{\log n}\right), 1 - \frac{1}{2}\exp\left(-2B\sqrt{\log n}\right)]$,

$$\left|\ln\left(p/(1-p)\right) - \ln\left(p'/(1-p')\right)\right| \leq 4\exp\left(2B\sqrt{\log n}\right)|p - p'|\,.$$

Putting things together. For the small L regime, we have with probability $1 - O(\frac{K}{n^8}) - O(K^2n^2\exp\left(-L^{0.99}\right))$, the following holds for all $k \in [K]$.

$$\begin{aligned}
\|\hat{\theta}^k - \theta^{*k}\|_2^2 &\leq \frac{\|\hat{\phi}^k - \phi^{*k}\|_F^2}{2n} \\
&= \frac{\sum_{i,j}(\hat{\phi}_{ij}^k - \phi_{ij}^{*k})^2}{2n} \\
&\lesssim \frac{\sum_{ij}\exp\left(4B\sqrt{\log n}\right)\cdot(\hat{P}_{ij}^k - P_{ij}^{*k})^2}{n} \\
&\lesssim \exp\left(4B\sqrt{\log n}\right) n \cdot \left(\frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}} + \frac{1}{e^{L^{0.99}}}\right)
\end{aligned}$$

For the large L regime, we have with probability $1 - O(\frac{K}{n^8}) - O(K^2n^2\exp\left(-n^\alpha\right))$, the following holds for all $k \in [K]$.

$$\begin{aligned}
\|\hat{\theta}^k - \theta^{*k}\|_2^2 &\leq \frac{\|\hat{\phi}^k - \phi^{*k}\|_F^2}{2n} \\
&= \frac{\sum_{i,j}(\hat{\phi}_{ij}^k - \phi_{ij}^{*k})^2}{2n} \\
&\lesssim \frac{\sum_{ij}\exp\left(4B\sqrt{\log n}\right)\cdot(\hat{P}_{ij}^k - P_{ij}^{*k})^2}{n} \\
&\lesssim \exp\left(4B\sqrt{\log n}\right) n \cdot \left(\frac{\sqrt{K}\sqrt{\log n}}{\sqrt{m}}\right)
\end{aligned}$$

By noting that

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_F \leq \sqrt{K}\cdot\max_k\|\theta^k - \theta^{*k}\|_2\,,$$

we obtain the bounds in the theorem statement. This finishes the proof. ■

Technical Remarks. As noted earlier, the $\exp\left(O(\sqrt{\log n})\right)$ in the Δ quantity is related to the dynamic gap which scales as $\sqrt{\log n}$ in our generative model. However, the reader might see that this term appears in the final bound as a consequence of a worst case analysis. For this reason, we are not able to obtain finite sample error bound for the very sparse regime $L \asymp \log n$ or $L = O(1)$. However, it might be possible to obtain a more refined lower bound for Δ that removes this term. With this, one could extend the theorem to the small L regime.

LSR versus Least Squares for Initialization. The reader might wonder why the spectral initialization does not use the LSR algorithm after the initial clustering step. This is a reasonable question since the algorithm is essentially a cluster-then-learn algorithm so the learning component can be delegated to the LSR algorithm. On the theoretical side, the least squares estimator presents an elegant theoretical analysis that can also be extended to more general RUMs other than PL. We propsoe the spectral initialization as a flexible algorithm for learning mixtures of general RUMs, not just mixtures of PLs. On the practical side, the least squares algorithm reduces the estimation problem to a constrained least-squares optimization problem which can be efficiently solved using off-the-shelf optimizers. Therefore, there are important reasons to consider applying the least squares estimator to the spectral initialization algorithm over the (unweighted) LSR algorithm. Experimentally, we observe that spectral initialization with LSR produces similar estimates to spectral initialization with the least squares estimator and the final estimates obtained by the EM algorithm are very similar.

C The Weighted LSR Algorithm

In this section, we will prove that the output of the weighted LSR algorithm is also the weighted MLE – Equation (6).

The Weighted Log-likelihood. The connection between the weighted ILSR algorithm and the weighted log-likelihood estimate stems from the generative model of the PL model as a sequence of choice. Namely, for a ranking π. The log-likelihood decomposes as

$$\log \mathbb{P}^{PL}(\pi \,|\, \theta) = \sum_{i=1}^{n-1} \log \left(\frac{e^{\theta_{\pi_i}}}{\sum_{j \geq i} e^{\theta_{\pi_j}}} \right).$$

Given a ranking dataset Π and sample weight vector q, the weighted log-likelihood is defined as

$$\mathcal{L}(\Pi, q, \theta) = \sum_{l=1}^{m} \sum_{i=1}^{n-1} q_l \cdot \log \left(\frac{e^{\theta_{\pi_i}}}{\sum_{j \geq i} e^{\theta_{\pi_j}}} \right).$$

Similarly to the analysis of the unweighted LSR algorithm in Maystre and Grossglauser [2015], we draw the connection between MLE and spectral ranking via the following reparametrization. Let $\hat{\theta}$ be the weighted MLE. Now define

$$\hat{p}_i = \frac{e^{\hat{\theta}_i}}{\sum_{k \in [n]} e^{\hat{\theta}_k}} \;\; i \in [n].$$

Naturally, $\hat{p} \in \mathcal{P}_n$ where $\mathcal{P}_n$ is the open probability symplex in n dimension. In terms of $\hat{p}$, the weighted log-likelihood is

$$\mathcal{L}(\Pi, q, \hat{p}) = \sum_{l=1}^{m} \sum_{i=1}^{n-1} q_l \cdot \log \left(\frac{\hat{p}_{\pi_i}}{\sum_{j \geq i} \hat{p}_{\pi_j}} \right).$$

Choice Breaking. Recall that the weighted ILSR algorithm first performs choice breaking on the permutations and the notion of choice breaking as a collection of choice enumerations composing the rankings. The weighted log-likelihood can be written in terms of the choice breaking as follows.

$$\mathcal{L}(\Pi, q, \hat{p}) = \sum_{(i, A, s) \in \mathcal{B}} q_s \cdot \log \left(\frac{\hat{p}_i}{\sum_{k \in A} \hat{p}_k} \right).$$

Focusing on a single index $i \in [n]$, the terms in the weighted log-likelihood that depends on $\hat{p}_i$ is

$$\mathcal{L}_i(\Pi, q, \hat{p}) = \sum_{(j, A, s) \in \mathcal{B} : i = j} q_s \cdot \log \left(\frac{\hat{p}_j}{\sum_{k \in A} \hat{p}_k} \right) + \sum_{(j, A, s) \in \mathcal{B} : i \neq j, i \in A} q_s \cdot \log \left(\frac{\hat{p}_j}{\sum_{k \in A} \hat{p}_k} \right).$$

Taking the derivative of this function with respect to $\hat{p}_i$ and setting the result to 0 gives

$$\frac{\partial \mathcal{L}_i(\Pi, q, \hat{p})}{\partial \hat{p}_i} = \sum_{(j,A,s)\in\mathcal{B}:i=j} q_s \cdot \left(\frac{1}{\hat{p}_i} - \frac{1}{\sum_{k\in A}\hat{p}_k}\right) + \sum_{(j,A,s)\in\mathcal{B}:i\neq j, i\in A} \left(-\frac{q_s}{\sum_{k\in A}\hat{p}_k}\right) = 0\,.$$

Multiplying both sides by $\hat{p}_i$ gives

$$\sum_{(j,A,s)\in\mathcal{B}:i=j} q_s \cdot \left(1 - \frac{\hat{p}_i}{\sum_{k\in A}\hat{p}_k}\right) + \sum_{(j,A,s)\in\mathcal{B}:i\neq j, i\in A} \left(-\frac{q_s\hat{p}_i}{\sum_{k\in A}\hat{p}_k}\right) = 0\,.$$

Rearranging the terms gives

$$\sum_{(j,A,s)\in\mathcal{B}:i=j} q_s \cdot \sum_{u\in A, u\neq i} \frac{\hat{p}_u}{\sum_{k\in A}\hat{p}_k} = \sum_{(j,A,s)\in\mathcal{B}:i\neq j, i\in A} \left(\frac{q_s\hat{p}_i}{\sum_{k\in A}\hat{p}_k}\right).$$

The reader may recognize that this equality is essentially

$$\sum_{j\neq i} \hat{p}_j \underbrace{\sum_{(i,A,s)\in\mathcal{B}_{i\succ j}} \frac{q_s}{\sum_{k\in A}\hat{p}_k}}_{M(\hat{p})_{ji}} = \sum_{j\neq i} p_i \underbrace{\sum_{(j,A,s)\in\mathcal{B}_{j\succ i}} \frac{q_s}{\sum_{k\in A}\hat{p}_k}}_{M(\hat{p})_{ij}},$$

where $M(\hat{p})_{ij}$ is defined in Equation (12). This is the stationarity condition for a Markov chain with stationary distribution $\hat{p}$ and state transition probabilities $M(\hat{p})_{ij}$'s. This system of equation also means that the weighted MLE $\hat{p}$ satisfies a notion of self-consistency. Namely,

$$\hat{p}^\top M(\hat{p}) = \hat{p}\,.$$

If we consider the iterative process of Algorithm 4, then $\hat{p}$ is a fixed point iterate. The rest of this section is devoted to proving that, starting from any initial distribution, the process converges to the weighted MLE $\hat{p}$.

Theorem 2. *The output of weighted LSR (Algorithm 4) is the maximum weighted log-likelihood estimate:*

$$\theta_q^{MLE} = \arg\max_\theta \sum_{l=1}^m q_l \cdot \log \mathbb{P}^{PL}(\pi_l \,|\, \theta)\,,$$

where $\mathbb{P}^{PL}(\pi_l\,|\theta)$ is shown in Equation (1).

Proof. We adapt our proof from the proof of Theorem 1 in Maystre and Grossglauser [2015]. We first prove a result analogous to Lemma 1 in that paper. Firstly, for a integer $t \geq 1$, define

$$T(p^\top) = p^\top M(p)$$

and

$$S_t(p^\top) = \underbrace{T(T(\ldots T(p)))}_{t \text{ times}} .$$

As shown previously, $\hat{p}$ satisfies $\hat{p}^\top = T(\hat{p}^\top)$. That is, $\hat{p}$ is a fixed point of the mapping T. If we can also show that T is a *contraction mapping*, i.e., $\|T(x) - T(y)\| \leq c \cdot \|x - y\| \, \forall x, y \in \mathcal{P}_n$ for some $c < 1$, then we could guarantee that

$$\lim_{t \to \infty} S_t(p^\top) = \hat{p}^\top \quad \forall p \in \mathcal{P}_n .$$

Towards this end, we need to analyze the Jacobian J of the transformation T. Firstly, note that $J \in \mathbb{R}^{n \times n}$. Recall that

$$M(p)_{ij} = \begin{cases} \epsilon \cdot \sum_{(i,A,s) \in \mathcal{B}_{j \succ i}} \frac{w_s}{\left(\sum_{k \in A} p_k\right)} & \text{if } i \neq j \\ 1 - \epsilon \cdot \sum_{(i,A,s) \in \mathcal{B}_{j \succ i}} \frac{w_s}{\left(\sum_{k \in A} p_k\right)} & \text{otherwise} \end{cases} .$$

By definition of the Jacobian, we have for an entry i, j where $i \neq j$ (an off diagonal entry),

$$[J(p^\top)]_{ij} = \left(\frac{\partial p^\top M(p)}{\partial p_i}\right)_j = \frac{\partial \left(p^\top M(p)\right)_j}{\partial p_i} = \frac{\partial \left(\sum_k p_k M(p)_{kj}\right)}{\partial p_i} .$$

We will now expand on this quantity.

$$\begin{aligned} [J(p^\top)]_{ij} &= \frac{\partial \left(\sum_k p_k M(p)_{kj}\right)}{\partial p_i} \\ &= \frac{\partial p_i M(p)_{ii}}{\partial p_i} + \sum_{k \neq i} \frac{\partial \left(p_k M(p)_{kj}\right)}{\partial p_i} \\ &= M(p)_{ij} + p_i \cdot \frac{\partial P_{ij}}{\partial p_i} + \sum_{k \neq i} \frac{\partial (p_k M(p)_{kj})}{\partial p_i} \\ &= M(p)_{ij} + p_i \cdot \frac{\partial P_{ij}}{\partial p_i} + \frac{\partial (p_j M(p)_{jj})}{\partial p_i} + \sum_{k \neq i,j} \frac{\partial (p_k M(p)_{kj})}{\partial p_i} \\ &= M(p)_{ij} + \frac{\partial (p_j M(p)_{jj})}{\partial p_i} + \sum_{k \neq j} p_k \cdot \frac{\partial M(p)_{kj}}{\partial p_i} \quad (*) . \end{aligned}$$

We will expand on the second and the third term.

$$
\begin{aligned}
&\frac{\partial(p_j M(p)_{jj})}{\partial p_i} + \sum_{k\neq j} p_k \cdot \frac{\partial M(p)_{kj}}{\partial p_i} \\
&= \sum_{k\neq j} p_k \cdot \epsilon \cdot \sum_{j,A,s\in\mathcal{B}_{j\succ k,i}} \frac{-w_s}{(\sum_{k'\in A} p_{k'})^2} - p_j \cdot \epsilon \cdot \sum_{k\neq j} \frac{\partial M(p)_{jk}}{\partial p_i} \\
&= \epsilon \cdot \sum_{j,A,s\in\mathcal{B}_{j\succ i}} \sum_{k\in A, k\neq j} \frac{-w_s p_k}{(\sum_{k'\in A} p_{k'})^2} + \epsilon \cdot \sum_{k\neq j} \sum_{k,A,s\in\mathcal{B}_{k\succ j,i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} \\
&= -\epsilon \cdot \sum_{j,A,s\in\mathcal{B}_{j\succ i}} \sum_{k\in A, k\neq j} w_s \cdot \left(\frac{1}{\sum_{k'\in A} p_{k'}} - \frac{p_j}{(\sum_{k'\in A} p_{k'})^2} \right) + \epsilon \cdot \sum_{k\neq j} \sum_{k,A,s\in\mathcal{B}_{k\succ j,i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} \\
&= \underbrace{-\epsilon \cdot \sum_{j,A,s\in\mathcal{B}_{j\succ i}} \frac{w_s}{\sum_{k'\in A} p_{k'}}}_{M_{ij}} + \epsilon \cdot \sum_{j,A,s\in\mathcal{B}_{j\succ i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} + \epsilon \cdot \sum_{k\neq j} \sum_{k,A,s\in\mathcal{B}_{k\succ j,i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} \\
&= M(p)_{ij} + \epsilon \cdot \sum_{k} \sum_{k,A,s\in\mathcal{B}_{k\succ j,i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} ,
\end{aligned}
$$

where, as an overload of notations, $\mathcal{B}_{j\succ j,i} = \mathcal{B}_{j\succ i}$. In the last step, we have combined the last two terms into one. Subtituting the above into the display $(*)$ gives

$$
[J(p^\top)]_{ij} = \epsilon \cdot \sum_{k} \sum_{k,A,s\in\mathcal{B}_{k\succ j,i}} \frac{w_s p_j}{(\sum_{k'\in A} p_{k'})^2} .
$$

Recall that this is just for the off-diagonal terms of the Jacobian. For the diagonal terms, note that

$$
\begin{aligned}
\sum_{j=1}^{n} [J(p)]_{ij} &= \sum_j M(p)_{ij} + \sum_j \sum_k p_k \cdot \frac{\partial M(p)_{kj}}{\partial p_i} \\
&= 1 + \sum_k p_k \sum_j \underbrace{\frac{\partial}{\partial p_i} \left(\sum_j M(p)_{kj} \right)}_{=0} = 1
\end{aligned}
$$

This means that the Jacobian matrix is a row-stochastic matrix (its rows sum to 1). Now recall the operator S_t defined earlier as the successive application of the operator T for t times. Following the same argument as in the proof of Theorem 1 of Maystre and Grossglauser [2015], for any initial distribution p, so long as p lies within the open probability simplex (its entries are strictly positive) there exists an interger t_0 such that $S_t(p)$ is entrywise positive for $t \geq t_0$. Note that by definition,

$$
S_t'(p) = (T^t(p))' = \prod_{o=0}^{t-1} T'(S_o(p)) .
$$

Since $T' = J$ has been shown to be a row stochastic matrix and $S_o(p)$ is entrywise positive, $S_t'(p)$ is also entrywise positive and the rows of $S_t'(p)$ all sum to 1. The

rest of the proof follows identically to the proof of Theorem 1 in Maystre and Grossglauser [2015]. ■

D Extra Experiments Discussions

Datasets Metadata. Table 1 summarizes the metadata for all the real-life datasets used in our experiments.

Dataset	m	n	Reference
APA	18723	5	Diaconis [1989]
Irish-Meath	64081	14	Mattei and Walsh [2013]
Irish-North	43942	12	Mattei and Walsh [2013]
Irish-West	29988	9	Mattei and Walsh [2013]
SUSHI	4997	10	Kamishima [2003]
ML-10M	71567	10681	Harper and Konstan [2015]

Table 1: Datasets metadata and references.

Data Pre-processing. For the three Irish election datasets that contains ties, we perform random tie breaking. The APA and SUSHI datasets contain no ties. For the ML-10M dataset, we first construct a user-item rating matrix from the ratings data. We then apply non-negative low rank matrix completion algorithm (with rank 30) using the open-source implementation found in Zitnik and Zupan [2012] to 'fill in' the missing entries. For each user, the ranking for the items is determined by the value of the filled in entries. To select the subset of items for use in the experiments, we select the items with the highest variance in user rankings. For the small scale real life datasets, we use all the available training rankings and Table 1 shows the performance for the algorithms when given all available training data. For variants of the ML-10M datasets, we use up to 14k rankings for training. For model selection, K is chosen from $[2, 3, 4, 5, 6, 7, 8, 9, 10]$ using Bayesian Information Criterion on the validation set. BIC is defined as $d \log(m_{\text{val}}) - 2 \log(L)$ where d is the number of parameters in the mixture model, m_{val} is the number of validation rankings and L is the likelihood evaluated on the validation set.

Some Notes on Implementations. We adopt the implementation of the MM algorithm from Maystre [2015]. To solve the optimization problems in the M-step in EM-CML and EM-GMM, we use scipy's built-in SLSQP solver. Please refer to our code in the supplementary materials for the implementation of all algorithms used in our experiments. The final class probability estimates are

taken as $\hat{\beta}^k \propto \sum_{l=1}^{m} q_l^k$ where q_l^k is the posterior class probability computed using the final estimate $\hat{\boldsymbol{\theta}}$.

Partial rankings. Firstly, note that the EM-LSR algorithm proposed in our paper can be decomposed into two steps. The first is an initialization procedure and the second is an EM refinement procedure. In the case of full rankings, initialization is delegated to a clustering-then-estimate approach where the clustering algorithm is spectral clustering. For partial rankings, we can replace spectral clustering by kernel clustering using semi-definite programming (SDP) Peng and Wei [2007]. We can use either the Kendall kernel or the Mallows kernel Jiao and Vert [2015] and their extensions to handle partial rankings Jiao and Vert [2015], Lomeli et al. [2018]. Once we have partitioned the rankings into clusters, the remaining of the spectral initialization proceeds in the same way. One can estimate the pairwise preference probabilities for each cluster and apply the least squares parameter estimation algorithm (Algorithm 2).

For the EM-procedure, note that the E-step can still be done efficiently because the PL likelihood for partial rankings can still be evaluated in closed form (see below). For the M-step, the weighted LSR algorithm can also handle partial rankings as it takes advantage of the decomposability of the Plackett-Luce likelihood function. The intuition for this comes from the interpretation of the generative process behind PL as a sequence of choice. Consider a partial ranking π, of s items where $s < n$, the universe size. The likelihood function for partial ranking is given as

$$\mathbb{P}(\pi \,|\, \theta) = \frac{e^{\theta_{\pi_1}}}{\sum_{j \geq 1} \theta_{\pi_j}} \cdots \frac{e^{\theta_{\pi_{s-1}}}}{e^{\theta_{\pi_{s-1}}} + e^{\theta_{\pi_s}}} \,.$$

Following the same construction as described in Section , one can perform choice breaking on partial rankings just like for full rankings and the algorithm requires little modifications. One can also inspect the proof of Theore 2 and see that the log-likelihood function decomposes into individual choice enumerations. Therefore, Theorem 2 also applies to the partial rankings setting as well.

Ties. As an example, we might observe a ranking

$$\pi = [\pi_1, \pi_2, \{\pi_3, \ldots, \pi_u\}, \pi_{u+1}, \ldots, \pi_s] \,,$$

where the items $\{\pi_3, \ldots, \pi_u\}$ are tied. To handle ties, we consider the size of the set of the tied items. If the set of tied items is small, we generate multiple copies of the original ranking, each with a possible permutation among the tied items, and divide the ranking weight by the number of such permutations. If the set of tied items is large, repeat this process but with a constant number of random permutations from the tied items. Alternatively we can also use a sampling algorithm recently proposed by Liu et al. [2019]. This sampling algorithm draws linear extensions (full rankings) consistent with given partial rankings and specified model parameters.

D.1 Additional Experiments

D.1.1 Random initialization vs spectral initialization

As mentioned in our main paper, we observe that spectral initialization outperforms random initialization. To help the reader see more clearly how much initialization helps, we augment Figure 1 in the main paper with results for random initialization and spectral initialization. The figure below shows how a random guess and the spectral initialization algorithm (no EM refinement) perform against the algorithms in the paper. We can see that the EM refinement step does result in a more accurate final estimate than spectral initialization alone. More importantly, spectral initialization provides a significantly better initial estimate than random initialization.

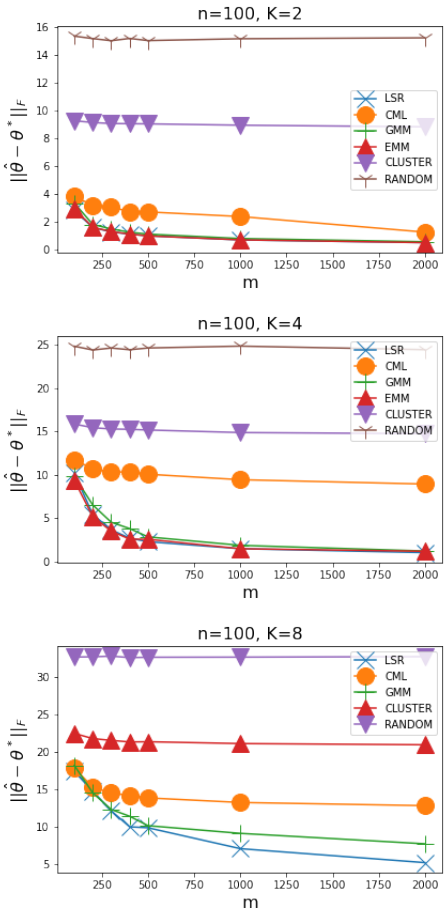


Figure 2: Synthetic datasets: The EM refinement procedure results in more accurate estimate than spectral initialization alone. Spectral initialization outperforms random initialization.

D.1.2 Comparison to other methods

Bayesian Methods. In the main paper we focus on parameter estimation and frequentist algorithms. Besides the frequentist (maximum likelihood) approach, some authors have also proposed Bayesian algorithms for PL parameter inference Guiver and Snelson [2009], Caron and Doucet [2012] for the classical PL model. For a mixture of PL models, we are only aware of the works by Mollica and Tardella [2017] and Caron et al. [2014]. The latter focus on clustering mixtures of Plackett Luce models while the former presented a learning algorithms for a finite mixture of PLs. Here we include extra experiments comparing EM-LSR and this Bayesian algorithm. To obtain final parameter estimates from the Bayesian model, we sample a 1000 samples from the posterior distribution and compute the average. **Smooth vs Stochastic M-step.** As noted in our main paper, Liu et al. [2019] used the unweighted LSR algorithm in their EM algorithm for learning mixtures of PL models. In order to incorporate unweighted LSR, they perform a probabilistic clustering in the M-step. The class posterior probabilities computed in the E-step are used to perform the clustering.

We perform additional experiments on synthetic datasets and ML-10M datasets comparing these three algorithms. Figure 3 shows the results on synthetic datasets and Figure 4 shows the results on ML-10M datasets. EM-LSR and the Bayesian algorithm have their own advantages over the other algorithm. EM-LSR is faster and requires little tuning. The Bayesian method, on the other hand, can be used to obtain uncertainty quantification as we learn a posterior distribution over the parameters $\boldsymbol{\theta}$ and produces slightly better estimates, especially in the data-poor regime. The tradeoff is of course speed. While the EM-LSR algorithm with a probabilistic clustering M-step is approximately faster by a constant factor than the original EM-LSR algorithm (both are orders of magnitude faster than the Bayesian method), it produces significantly worse estimates. We suspect that the noise introduced by the stochastic M-step causes the trajectory of the EM algorithm to end up in a bad local optimum.

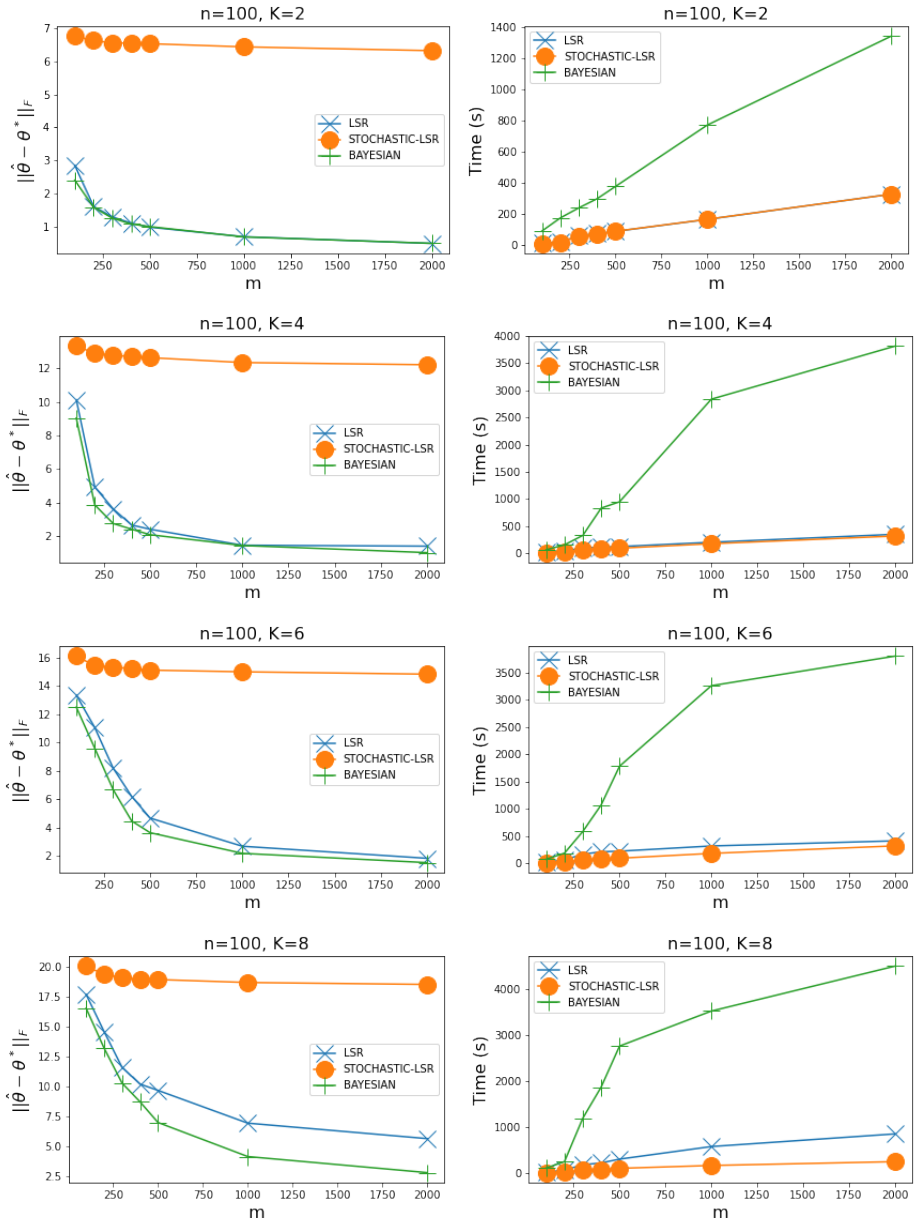


Figure 3: Synthetic datasets: The Bayesian method produces slightly better estimates than smooth EM-LSR but is much slower than the EM-LSR algorithms. Smooth EM-LSR is significantly more accurate than stochastic EM-LSR.

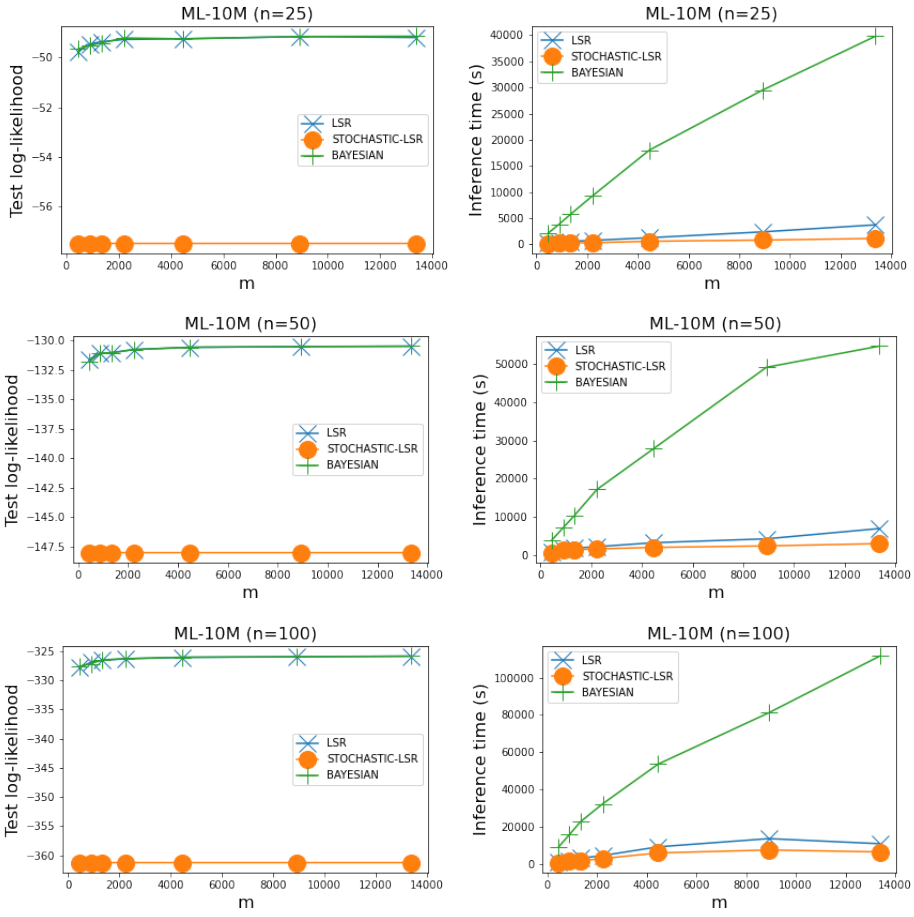


Figure 4: ML-10M datasets: On real datasets, soft EM-LSR is more accurate than stochastic EM-LSR. Smooth EM-LSR is similar in accuracy to the Bayesian method but is orders of magnitude faster.

References

Francois Caron and Arnaud Doucet. Efficient bayesian inference for generalized bradley–terry models. *Journal of Computational and Graphical Statistics*, 21 (1):174–196, 2012.

Francois Caron, Yee Whye Teh, and Thomas Brendan Murphy. Bayesian nonparametric plackett–luce models for the analysis of preferences for college degree programmes. *The Annals of Applied Statistics*, 8(2):1145–1181, 2014.

Persi Diaconis. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, pages 949–979, 1989.

John Guiver and Edward Snelson. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, pages 377–384, 2009.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Yunlong Jiao and Jean-Philippe Vert. The kendall and mallows kernels for permutations. In *International Conference on Machine Learning*, pages 1935–1944. PMLR, 2015.

Toshihiro Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588, 2003.

Jing Lei and Alessandro Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

Ao Liu, Zhibing Zhao, Chao Liao, Pinyan Lu, and Lirong Xia. Learning plackett-luce mixtures from partial preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4328–4335, 2019.

Maria Lomeli, Mark Rowland, Arthur Gretton, and Zoubin Ghahramani. Antithetic and monte carlo kernel estimators for partial rankings. *arXiv preprint arXiv:1807.00400*, 2018.

Nicholas Mattei and Toby Walsh. Preflib: A library for preferences http://www.preflib. org. In *International Conference on Algorithmic Decision Theory*, pages 259–270. Springer, 2013.

Lucas Maystre. Choix. https://github.com/lucasmaystre/choix, 2015.

Lucas Maystre and Matthias Grossglauser. Fast and accurate inference of plackett–luce models. *Advances in neural information processing systems*, 28, 2015.

Cristina Mollica and Luca Tardella. Bayesian plackett–luce mixture models for partially ranked data. *Psychometrika*, 82(2):442–458, 2017.

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205, 2007.

Devavrat Shah and Dogyoon Song. Learning rums: Reducing mixture to single component via pca. *arXiv preprint arXiv:1812.11917*, 2018.

Gilbert Strang, Gilbert Strang, Gilbert Strang, and Gilbert Strang. *Introduction to linear algebra*, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.

Louis L Thurstone. A law of comparative judgment. In *Scaling*, pages 81–92. Routledge, 1927.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Anderson Y Zhang and Harrison H Zhou. Leave-one-out singular subspace perturbation analysis for spectral clustering. *arXiv preprint arXiv:2205.14855*, 2022.

Marinka Zitnik and Blaz Zupan. Nimfa: A python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, 13:849–853, 2012.