The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Mixtures of Random Utility Models

Zhibing Zhao, Tristan Villamil, Lirong Xia
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY, USA
{zhaoz6, villat2}@rpi.edu, xial@cs.rpi.edu

Abstract

We tackle the problem of identifiability and efficient learning of mixtures of Random Utility Models (RUMs). We show that when the PDFs of utility distributions are symmetric, the mixture of k RUMs (denoted by k-RUM) is not identifiable when the number of alternatives m is no more than $2k-1$. On the other hand, when $m \geq \max\{4k-2, 6\}$, any k-RUM is *generically* identifiable. We then propose three algorithms for learning mixtures of RUMs: an EM-based algorithm, which we call *E-GMM*, a direct generalized-method-of-moments (GMM) algorithm, and a *sandwich (GMM-E-GMM)* algorithm that combines the other two. Experiments on synthetic data show that the sandwich algorithm achieves the highest statistical efficiency and GMM is the most computationally efficient. Experiments on real-world data at Preflib show that Gaussian k-RUMs provide better fitness than a single Gaussian RUM, the Plackett-Luce model, and mixtures of Plackett-Luce models w.r.t. commonly-used model fitness criteria. To the best of our knowledge, this is the first work on learning mixtures of general RUMs.

Introduction

In *rank aggregation*, the goal is to aggregate rank data to make an optimal decision, where each data point is a linear order over a set of alternatives. Rank aggregation has many applications. For example, in political elections, agents cast votes to elect a president; in information retrieval, rankings over documents are combined into a list; in crowdsourcing, crowd workers sometimes give rankings as answers, which are aggregated to estimate the correct ranking.

In this paper, we take a machine learning approach towards rank aggregation, by addressing the problem of efficiently learning mixtures of *Random Utility Models (RUMs)*. RUMs are a widely applied statistical model for human behaviors (Thurstone 1927). In a single, non-mixture RUM, each agent samples a utility for each alternative independently and reports the ranking over alternatives by sorting their utilities. A special case of RUMs is the *Plackett-Luce model* (Plackett 1975; Luce 1959), which can be seen as the extension of multinomial logistic regression to rank data. General RUMs can fit data better than Plackett-Luce models and thus provide more accurate predictions (Azari Soufiani, Parkes, and Xia 2012). Unfortunately, unlike the Placket-Luce model, general RUMs are hard to learn due to the lack of closed-form formula of the likelihood function. One prominent example is Thurstone's case V model (Thurstone 1927), where the utility distributions are Gaussian and can be seen as an extension of multinomial probit regression to rank data. See Related Work for more discussions.

A mixture of k RUM models, denoted by k-RUM, combines k RUM components via the *mixing coefficients* $\vec{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_k]$. Given a k-RUM, a ranking can be generated as follows: (i) the rth component is selected with probability α_r; (ii) a ranking is generated from the rth RUM component. Like other mixture models, k-RUMs can potentially provide better fitness and can be used for clustering.

We are not aware of previous work on learning mixtures of general RUMs. There are some recent works on mixtures of Plackett-Luce models (Gormley and Murphy 2008; Mollica and Tardella 2016; Tkachenko and Lauw 2016; Zhao, Piech, and Xia 2016), which are special k-RUMs.

The motivation of our work on learning mixtures of general RUMs is that general k-RUMs outperforms a single RUM, the Plackett-Luce model, and mixtures of Plackett-Luce models w.r.t. various commonly-used model fitting criteria, as we will show later in the paper based on experiments on real-world data. This means that general k-RUMs often improve predictions and decisions compared to the state-of-the-art models.

In this paper, we address the following two fundamental questions on learning mixtures of RUMs. *Is a k-RUM learnable? Can we design efficient algorithms for learning k-RUMs in general?*

Our Contributions

In this paper, we measure learnability by "*identifiability*", which means different parameters correspond to different distributions of rank data. Identifiability is important to mixture models for parameter estimation and clustering. The identifiability of k-RUMs depends on the number of components k in the mixture model and the number of alternatives m, as we show in the following two theorems.

Theorem 1. *Let $\mathcal{M}$ be any symmetric RUM from the location family. When $m \leq 2k-1$, k-RUM$_{\mathcal{M}}$ over m alterna-*

tives is non-identifiable[1].

Here k-RUM$_{\mathcal{M}}$ denotes the mixture of k RUMs, each of which is chosen from the RUM model $\mathcal{M}$. As a corollary, the k-mixture of Thurstone's original case V model (with Gaussian distributions) is non-identifiable when $m \leq 2k-1$. Our second theorem proves *generic identifiability*. k-RUM$_{\mathcal{M}}$ is *generically* identifiable, if the Lebesgue measure of non-identifiable parameters is 0.

Theorem 2. *For any RUM$_{\mathcal{M}}$ where all utility distributions have support $(-\infty, \infty)$, when $m \geq \max\{4k-2, 6\}$, k-RUM$_{\mathcal{M}}$ over m alternatives is generically identifiable.*

We propose three algorithms to learn k-RUMs. The first one is an EM-based algorithm (Dempster, Laird, and Rubin 1977), which we call E-GMM. It works as follows: the E-step is standard, calculating the membership of each ranking in the data w.r.t. the k components based on the likelihood ratios. In the M step, we adopt the *generalized-method-of-moments (GMM)* algorithm proposed by Azari Soufiani, Parkes, and Xia (2014) to estimate parameters for each RUM component.

Our second algorithm is a direct GMM algorithm (Hansen 1982) that tries to match the moments for k-RUM. We prove that under mild conditions the algorithm is consistent. That is, it converges to the ground truth with probability going to 1 as the number of independently generated rankings approaches infinity.

Our third algorithm combines the advantages of the E-GMM algorithm and the GMM algorithm. We first run GMM, use the output as the initial values to run E-GMM. The algorithm is thus called the *sandwich* algorithm (GMM-E-GMM). We note that the two GMMs in the sandwich algorithm are different.

Experiments on synthetic data show that the sandwich algorithm is the most statistically efficient and the GMM algorithm is the fastest. Observing that the E-GMM algorithm is not as good as the sandwich algorithm w.r.t. both statistical efficiency and computational efficiency, we don't show the results of the E-GMM algorithm in this paper.

Experiments on real-world Preflib data (Mattei and Walsh 2013) show that k-RUMs provide better model fitness than a single RUM, the Plackett-Luce model, and mixtures of Plackett-Luce models w.r.t. commonly-used model fitness criteria, including Akaike Information Criterion (AIC) (Akaike 1974), corrected AIC for finite samples (AICc) (Hurvich and Tsai 1989), and Bayesian Information Criterion (BIC) (Schwarz 1978). We note that such comparisons and conclusions are enabled by our algorithms on learning general k-RUMs, which are the first of their kind.

Related Work and Discussions

Our (non)-identifiability theorems are the first ones for mixtures of general RUMs. Recently Zhao, Piech, and Xia (2016) characterized identifiability of mixtures of Plackett-Luce models. We focus on a different class of mixtures of RUMs—those whose utility distributions have symmetric PDFs, including Thurstone's case V model. In order to prove the (non)-identifiability theorem, we adopted novel techniques by avoiding direct calculation of the likelihood. This is the major difference between our proofs and the proofs of Zhao, Piech, and Xia (2016) for mixtures of Plackett-Luce models, where closed-form formulas for the likelihood function are available. We emphasize that our non-identifiability theorem does not apply to mixtures of Plackett-Luce models because the utility distributions in the Plackett-Luce model (Gumbel distributions) are not symmetric.

Our work also makes a number of algorithmic contributions to the literature of rank aggregation, sometimes known as *learning to rank* (Liu 2011). In particular, GMM-based techniques for learning a single (non-mixture) RUM, including the Plackett-Luce model, have been extensively investigated (Negahban, Oh, and Shah 2012; Azari Soufiani et al. 2013; Azari Soufiani, Parkes, and Xia 2014; Chen and Suh 2015; Khetan and Oh 2016b; 2016a). Learning algorithms have also been proposed for mixtures of Plackett-Luce models, including EM-based algorithms (Gormley and Murphy 2008; Mollica and Tardella 2016; Tkachenko and Lauw 2016) and a generalized method of moments algorithm (Zhao, Piech, and Xia 2016). To the best of our knowledge, our algorithms for learning k-RUMs are quite general and are the first algorithms for learning mixtures of RUMs beyond mixtures of Plackett-Luce models.

Our E-GMM is inspired by the EMM algorithm for learning mixtures of Plackett-Luce models proposed by Gormley and Murphy (2008). However, the EMM cannot be easily applied to learn general k-RUMs, because EMM uses a *Minorize-Maximization (MM)* algorithm to estimate parameters of each Plackett-Luce component, but no MM algorithm is known for general RUMs. To address this challenge, we adopt a GMM by Azari Soufiani, Parkes, and Xia (2014) in the M step. Our GMM algorithm is inspired by the GMMs in the previous work discussed above but we use a different set of moment conditions.

Recently tensor-decomposition techniques have been explored to learn models with latent variables, see for example the work by Anandkumar et al. (2014). However, such techniques cannot be easily applied to learn RUMs and their mixtures. In the proof of Theorem 2 we shed some light on a potential algorithm, which requires multiple tensor decompositions and each component is only labeled by its mixing probability. Designing computationally tractable tensor-decomposition algorithms is a promising direction for future work.

Preliminaries

Let $\mathcal{A} = \{a_i | i = 1, 2, \cdots, m\}$ denote a set of m alternatives. Let $\mathcal{L}(\mathcal{A})$ be the set of linear orders (full rankings) over $\mathcal{A}$, which are transitive, antisymmetric and total binary relations. Let $P = \{V_1, V_2, \cdots, V_n\}$ denote the data (also called a *preference profile*), where for all $j \leq n$, $V_j \in \mathcal{L}(\mathcal{A})$.

Definition 1 *(Random Utility Model (RUM)). Given $m \geq 2$ alternatives, a random utility model $\mathcal{M}$ associates each al-*

[1] All identifiability results in this paper hold modulo label switching, which means that if we label the components differently, the parameter is treated the same.

ternative a_i with a utility distribution. The parameter space is $\Theta = \{\vec{\theta} = \{\vec{\theta}_i | i = 1, 2, \cdots, m\}\}$, where $\vec{\theta}_i$ is the parameter of the utility distribution of a_i. The sample space is $\mathcal{S} = \mathcal{L}(\mathcal{A})^n$. An agent's ranking is generated in the following steps: first, the agent samples a random utility U_i for each alternative independently from $\mu_i(\cdot|\vec{\theta}_i)$; second, she ranks the alternatives w.r.t. these utilities. Formally, given a parameter $\vec{\theta}$, the probability of ranking $V = a_{i_1} \succ a_{i_2} \succ \cdots \succ a_{i_m}$ is

$$\Pr_{\mathcal{M}}(V|\vec{\theta}) = \int_{-\infty}^{\infty}\int_{x_{i_m}}^{\infty}\cdots\int_{x_{i_2}}^{\infty} \mu_{i_m}(x_{i_m}|\vec{\theta}_{i_m}) \cdots \mu_{i_1}(x_{i_1}|\vec{\theta}_{i_1}) dx_{i_1} dx_{i_2} \cdots dx_{i_m}$$

We assume that rankings are generated i.i.d. Therefore, given a preference profile P and $\vec{\theta} \in \Theta$, we have $\Pr_{\mathcal{M}}(P|\vec{\theta}) = \prod_{j=1}^{n} \Pr_{\mathcal{M}}(V_j|\vec{\theta})$.

In this paper, we focus on the *location family*, where each utility distribution μ_i is only parameterized by its mean, denoted by θ_i. In other words, the shapes of the utility distributions are fixed. We note that shifting the means of all alternatives simultaneously by the same distance will not affect the distribution of the rankings. To eliminate this problem, **we require** $\sum_{i=1}^{m} \theta_i = 0$. We further say that an RUM is *symmetric* if the PDF of each utility distribution is symmetric around its mean.

Definition 2 *(k-RUM$_{\mathcal{M}}$). Given an RUM $\mathcal{M}$ within the location family, for any $k \in \mathbb{N}^+$ and $m \geq 2$, we define the k-*RUM$_{\mathcal{M}}$ *as follows. The sample space is $\mathcal{S} = \mathcal{L}(\mathcal{A})^n$. The parameter space has two parts. The first part is the* mixing coefficients $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ *where for all $r \leq k$, $\alpha_r \geq 0$, and $\sum_{r=1}^{k} \alpha_r = 1$. The second part is $(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(k)})$, where $\vec{\theta}^{(r)}$ is the parameter of the r-th random utility component. All components come from the same class of RUM $\mathcal{M}$, i.e. the shape of the utility distribution of any alternative is fixed. The probability of a ranking V is $\Pr_{k\text{-}RUM_{\mathcal{M}}}(V|\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \Pr_{\mathcal{M}}(V|\vec{\theta}^{(r)})$, where $\Pr_{\mathcal{M}}(V|\vec{\theta}^{(r)})$ is the probability of generating V in the r-th RUM.*

Example 1 *Consider a 2-RUM over 3 alternatives. The mixing coefficients are $\vec{\alpha} = (0.3, 0.7)$. Let all utility distributions be Gaussian distributions with standard deviation 1. The two RUM components are parameterized by $\vec{\theta}^{(1)} = (-3, -1, 4)$ and $\vec{\theta}^{(2)} = (5, -2, -3)$, respectively. Taking the ranking $a_1 \succ a_3 \succ a_2$ as an example, we have* $\Pr(a_1 \succ a_3 \succ a_2) = 0.3 \times \int_{-\infty}^{\infty}\int_{x_2}^{\infty}\int_{x_3}^{\infty} \phi(x_2+1)\phi(x_3-4)\phi(x_1+3)dx_1dx_3dx_2 + 0.7 \times \int_{-\infty}^{\infty}\int_{x_2}^{\infty}\int_{x_3}^{\infty} \phi(x_2+2)\phi(x_3+3)\phi(x_1-5)dx_1dx_3dx_2$, *where $\phi(x)$ is the PDF of the standard Gaussian distribution.*

Identifiability of k-RUM$_{\mathcal{M}}$

We first recall the identifiability of a statistical model.

Definition 3 *(Identifiability) Let $\mathcal{M} = \{\Pr(\cdot|\vec{\theta}) : \vec{\theta} \in \Theta\}$ be a statistical model. $\mathcal{M}$ is identifiable if for all $\vec{\theta}, \vec{\gamma} \in \Theta$, we have* $\Pr(\cdot|\vec{\theta}) = \Pr(\cdot|\vec{\gamma}) \Longrightarrow \vec{\theta} = \vec{\gamma}$.

We note that single RUMs are generally identifiable and can be learned using an MC-EM algorithm (Azari Soufiani, Parkes, and Xia 2012) or a GMM algorithm (Azari Soufiani, Parkes, and Xia 2014). However, identifiability of single RUMs does not imply identifiability of mixtures of RUMs.

To eliminate the label switching problem (Redner and Walker 1984), we say that k-RUM$_{\mathcal{M}}$ is identifiable if there do not exist (1) $1 \leq k_1, k_2 \leq k$, *non-degenerate* $\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \cdots, \vec{\theta}^{(k_1)}$, $\vec{\gamma}^{(1)}, \vec{\gamma}^{(2)}, \cdots, \vec{\gamma}^{(k_2)}$, meaning that these $k_1 + k_2$ vectors are pairwise different; (2) all strictly positive mixing coefficients $(\alpha_1^{(1)}, \ldots, \alpha_{k_1}^{(1)})$ and $(\alpha_1^{(2)}, \ldots, \alpha_{k_2}^{(2)})$, so that for any ranking V in P we have:

$$\sum_{r=1}^{k_1} \alpha_r^{(1)} \Pr_{\mathcal{M}}(V|\vec{\theta}^{(r)}) = \sum_{r=1}^{k_2} \alpha_r^{(2)} \Pr_{\mathcal{M}}(V|\vec{\gamma}^{(r)})$$

A statistical model is *generically identifiable*, if the Lebesgue measure of parameters that does not satisfy the condition in Definition 3 is zero in the parameter space.

Theorem 1 *Let $\mathcal{M}$ be any symmetric RUM from the location family. When $m \leq 2k - 1$, k-RUM$_{\mathcal{M}}$ over m alternatives is non-identifiable.*

Proof: The proof is constructive. We prove the case where $m = 2k - 1$. The proof for cases where $m < 2k - 1$ is similar. For any k and $m = 2k-1$, we will define non-degenerate $\vec{\theta}^{(1)}, \ldots, \vec{\theta}^{(k)}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)}$ with mixing probabilities $\vec{\alpha} = [\alpha_1, \ldots, \alpha_k]^T$ and $\vec{\beta} = [\beta_1, \ldots, \beta_k]^T$, respectively. We let $\theta_1^{(1)}, \ldots, \theta_1^{(k)}, \gamma_1^{(1)}, \ldots, \gamma_1^{(k)}$ be $2k$ pairwise different numbers where for all $r = 1, \ldots, k$, $\theta_1^{(r)} + \gamma_1^{(r)} = 0$. For any $\vec{\theta}^{(r)}$, we let $\theta_2^{(r)} = \ldots = \theta_m^{(r)} = -\frac{\theta_1^{(r)}}{m-1}$ s.t. $\sum_{i=1}^{m} \theta_i^{(r)} = 0$. $\vec{\gamma}^{(r)}$'s are defined similarly.

Because the parameters for $a_2, \ldots, a_m$ are equal in each RUM component, the distribution over data in each RUM component can then be represented compactly using m events instead of $m!$ rankings, i.e. a_1 at the first position, second position, etc. For convenience we define $\mathbf{F}_\theta$ as

$$\mathbf{F}_\theta = \begin{bmatrix} \Pr(a_1 \text{ top}|\vec{\theta}^{(1)}) & \cdots & \Pr(a_1 \text{ top}|\vec{\theta}^{(k)}) \\ \vdots & \ddots & \vdots \\ \Pr(a_1 \text{ bottom}|\vec{\theta}^{(1)}) & \cdots & \Pr(a_1 \text{ bottom}|\vec{\theta}^{(k)}) \end{bmatrix}.$$

$\mathbf{F}_\gamma$ can be defined similarly. We will prove that there exist positive $\vec{\alpha}$ and $\vec{\beta}$ s.t. $\mathbf{F}_\theta \cdot \vec{\alpha} = \mathbf{F}_\gamma \cdot \vec{\beta}$, which will prove non-identifiability.

Consider the matrix $\Delta\mathbf{F} = \mathbf{F}_\theta - \mathbf{F}_\gamma$. For all $1 \leq r \leq k$, the rth column of $\Delta\mathbf{F}$ is the following:

$$\Delta\mathbf{F}^{(r)} = \begin{bmatrix} \Pr(a_1 \text{ top}|\vec{\theta}^{(r)}) - \Pr(a_1 \text{ top}|\vec{\gamma}^{(r)}) \\ \vdots \\ \Pr(a_1 \text{ bottom}|\vec{\theta}^{(r)}) - \Pr(a_1 \text{ bottom}|\vec{\gamma}^{(r)}) \end{bmatrix}.$$

Because the utility distributions of all alternatives are symmetric, we have

$$\Pr(a_1 \text{ position } i|\vec{\theta}^{(r)}) = \Pr(a_1 \text{ postion } m - i + 1|\vec{\gamma}^{(r)}).$$

Therefore, the first element of $\Delta\mathbf{F}^{(r)}$ is exactly the same as the last element of it except for a negative sign. The same holds for the second element and the last but one element, etc. The center element (the kth element) is zero. Since this holds for all r, the last $k-1$ rows of $\Delta\mathbf{F}$ are in the reversed order of the negative top $k-1$ rows, while the center row consists of only zeros. So the rank of $\Delta\mathbf{F}$ is at most $k-1$. Since $\Delta\mathbf{F}$ has k rows, there exists a nonzero $\vec{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_k]^\top$ s.t. $\Delta\mathbf{F} \cdot \vec{\lambda} = 0$. Elementwise we have $\forall i$: $\sum_{r=1}^{k} \lambda_r \Pr(a_1 \text{ position } i | \vec{\theta}^{(r)}) = \sum_{r=1}^{k} \lambda_r \Pr(a_1 \text{ position } i | \vec{\gamma}^{(r)})$.

If the entries in $\vec{\lambda}$ are all nonnegative, we let $\vec{\alpha} = \vec{\beta} = \vec{\lambda}$, and the proof is done. If there are negative elements in λ, we can switch the corresponding components $\vec{\theta}^{(r)}$ and $\vec{\gamma}^{(r)}$ and flip the sign of λ_r until all elements in $\vec{\lambda}$ are nonnegative. This finishes the proof. ■

We conjecture that this bound is tight since the counterexample has the fewest number of moments, which means the parameter is the most likely to be under-constrained.

Theorem 2 *For any RUM$_\mathcal{M}$ where all utility distributions have support $(-\infty, \infty)$, when $m \geq \max\{4k-2, 6\}$, k-RUM$_\mathcal{M}$ over m alternatives is generically identifiable.*

Proof: We will focus on the parameters whose mixing coefficients (entries of $\vec{\alpha}$) are pairwise different. Formally, for any $r_1, r_2 \in \{1, \ldots, k\}$,

$$\alpha_{r_1} = \alpha_{r_2} \implies r_1 = r_2. \tag{1}$$

Such parameters have Lebesgue measure 1 because the parameters with any pair of identical mixing coefficients are in a lower dimensional space. The generic identifiability will be proved by analyzing the uniqueness of tensor decompositions. We will construct a rank-one tensor $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$ to represent the rth RUM component. Then the k-RUM$_\mathcal{M}$ can be represented by the weighted average of tensors of its components, i.e. $\mathbf{T}(\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \mathbf{T}^{(r)}(\vec{\theta}^{(r)})$. We now provide a set of two sufficient conditions for $\vec{\theta} = (\vec{\alpha}, \vec{\theta}^{(1)}, \ldots, \vec{\theta}^{(k)})$ to be identifiable, and then prove that both hold generically.

Condition 1. For every $\vec{\gamma} = (\vec{\beta}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)})$, where $\vec{\beta} \neq \vec{\alpha}$ modulo label switching (the set of entries in $\vec{\beta}$ is different from the set of entries in $\vec{\alpha}$), we have $\mathbf{T}(\vec{\gamma}) \neq \mathbf{T}(\vec{\theta})$.

Condition 2. For every $\vec{\gamma} = (\vec{\beta}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)})$ that is different from $\vec{\theta}$ and $\vec{\beta} = \vec{\alpha}$, we have $\Pr_{k\text{-RUM}_\mathcal{M}}(\cdot|\vec{\gamma}) \neq \Pr_{k\text{-RUM}_\mathcal{M}}(\cdot|\vec{\theta})$.

It follows from the definition of identifiability that if both conditions hold, then $\vec{\theta}$ is identifiable.

Condition 1 generically holds. We first show that $\mathbf{T}(\vec{\theta})$ has a unique decomposition generically, then prove that the uniqueness of decomposition implies Condition 1.

In the rest of the proof we assume that m is even. The theorem can be proved similarly for odd m's. To construct the rank-one tensor $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$, we partition the set of alternatives into $m/2$ subsets. $S_1 = \{a_1, a_2\}, S_2 = \{a_3, a_4\}, \ldots, S_{m/2} = \{a_{m-1}, a_m\}$. We define the $\frac{m}{2}$-dimensional rank-one tensor $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$ for the rth RUM$_\mathcal{M}$ component as follows. Let the qth coordinate of $\mathbf{T}_r$ be the probabilities for the pairwise comparison between the two alternatives in S_q. More precisely, for any $1 \leq q \leq m/2$ and any rankings $V_q \in \mathcal{L}(S_q)$, the qth coordinate of $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$, denoted by $\mathbf{p}_q^{(r)}$, is the following:

$$\mathbf{p}_q^{(r)} = \left[\Pr(a_{2q-1} \succ a_{2q} | \vec{\theta}^{(r)}), \Pr(a_{2q} \succ a_{2q-1} | \vec{\theta}^{(r)})\right]^\top \tag{2}$$

According to Lemma 3 of (Zhao, Piech, and Xia 2016)[2], we have $\Pr_\mathcal{M}(V_1, V_2, \ldots, V_{m/2} | \vec{\theta}^{(r)}) = \prod_{q=1}^{m/2} \Pr_\mathcal{M}(V_q | \vec{\theta}^{(r)})$, where for all $q = 1, \ldots, m/2$, $V_q \in \mathcal{L}(S_q)$. It follows that $\mathbf{T}^{(r)}(\vec{\theta}^{(r)}) = \otimes_{q=1}^{m/2} \mathbf{p}_q^{(r)}$. The tensor for the mixture model can be written as $\mathbf{T}(\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \mathbf{T}^{(r)}(\vec{\theta}^{(r)})$. To analyze the uniqueness of tensor decomposition of $\mathbf{T}(\vec{\theta})$, we investigate the *Kruskal's rank* of each matrix $\mathbf{P}_q$ defined as $\mathbf{P}_q = \left[\mathbf{p}_q^{(1)}, \mathbf{p}_q^{(2)}, \cdots, \mathbf{p}_q^{(k)}\right]$. The Kruskal's rank of a matrix is the largest number K such that every set of K columns in the matrix is linearly independent. Now we will prove that the Kruskal rank of $\mathbf{P}_q$, denoted by K_q, is generically 2, i.e. any two columns of $\mathbf{P}_q$ are generically independent. It is not hard to see that the two entries of $\mathbf{p}_q^{(r)}$ sum up to 1. So the only case where $\mathbf{p}_q^{(r_1)}$ and $\mathbf{p}_q^{(r_2)}$ are linearly dependent is that they are exactly the same. According to Proposition 1 in (Azari Soufiani, Parkes, and Xia 2014), $\Pr(a_{2q-1} \succ a_{2q} | \vec{\theta}^{(r)})$ monotonically increases as $\theta_{2q-1}^{(r)} - \theta_{2q}^{(r)}$ increases because all utility distributions have support $(-\infty, \infty)$. Therefore, $\mathbf{p}_q^{(r_1)} = \mathbf{p}_q^{(r_2)}$ if and only if $\theta_{2q-1}^{(r_1)} - \theta_{2q}^{(r_1)} = \theta_{2q-1}^{(r_2)} - \theta_{2q}^{(r_2)}$, which has Lebesgue measure 0. Thus, for all $q \leq m/2$, $K_q = 2$ generically holds. So $\sum_{q=1}^{m/2} K_q = m/2 \times 2 = m$ generically holds. Because $k \leq \frac{m+2}{4}$, we have $\sum_{q=1}^{m/2} K_q = 2k + m/2 - 1$. Sidiropoulos and Bro (2000) proved that, for any N-way tensor, if $\sum_{q=1}^{N} K_q \geq 2k + N - 1$, then the tensor decomposition is unique. This is true in our case since $N = m/2$.

We now prove that uniqueness of decomposition of $\mathbf{T}(\vec{\theta})$ implies Condition 1. Suppose for the purpose of contradiction, the decomposition of $\mathbf{T}(\vec{\theta})$ is unique but Condition 1 does not hold. This means that there exists $\vec{\gamma} = (\vec{\beta}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)})$ where $\vec{\beta}$ is not equal to $\vec{\alpha}$ modulo label switching, s.t. $\mathbf{T}(\vec{\gamma}) \neq \mathbf{T}(\vec{\theta})$. Because components in $\vec{\alpha}$ are pairwise different, there exists $r_1 \leq k$ such that for all $r_2 = 1, \ldots, k$, $\alpha_{r_1} \neq \beta_{r_2}$, while $\mathbf{T}(\vec{\theta}) = \mathbf{T}(\vec{\gamma}) = \mathbf{T}$. We will show that for any $r_2 = 1, \ldots, k$, we have $\alpha_{r_1} \mathbf{T}^{(r_1)}(\vec{\theta}^{(r_1)}) \neq \beta_{r_2} \mathbf{T}^{(r_2)}(\vec{\gamma}^{(r_2)})$, which contradicts the unique decomposition of $\mathbf{T}$. Recall that for any r, the sum

[2] This lemma proved the independence of two rankings with mutually exclusive alternatives, which can be easily extended to multiple rankings.

of the two entries of $\mathbf{p}_q^{(r)}$ is 1. It is not hard to see the sum of all entries of $\mathbf{T}^{(r)}(\cdot)$ is also one. So the sums of all entries of $\alpha_{r_1}\mathbf{T}^{(r_1)}(\vec{\theta}^{(r_1)})$ and $\beta_{r_2}\mathbf{T}^{(r_2)}(\vec{\gamma}^{(r_2)})$ are α_{r_1} and β_{r_2} respectively. We recall that for all r_2, $\alpha_{r_1} \neq \beta_{r_2}$ holds. Therefore, $\alpha_{r_1}\mathbf{T}^{(r_1)}(\vec{\theta}^{(r_1)}) \neq \beta_{r_2}\mathbf{T}^{(r_2)}(\vec{\gamma}^{(r_2)})$ holds for all r_2, which is a contradiction. This finishes the proof that **Condition 1 generically holds**.

Condition 2 generically holds. Unfortunately the tensor decomposition technique used for Condition 1 no longer works for Condition 2. Here is a counterexample. Given $\vec{\theta} = (\vec{\alpha}, \vec{\theta}^{(1)}, \ldots, \vec{\theta}^{(k)})$, we can construct $\vec{\gamma} = (\vec{\alpha}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)})$ s.t. $\mathbf{T}(\vec{\gamma}) = \mathbf{T}(\vec{\theta})$ in the following way. For any r, we let $\gamma_i^{(r)} = \theta_i^{(r)} + 1$ for $i = 1, 2$, $\gamma_i^{(r)} = \theta_i^{(r)} - 1$ for $i = 3, 4$ and $\gamma_i^{(r)} = \theta_i^{(r)}$ for $i = 5, \ldots, m$. The resulting tensor $\mathbf{T}(\vec{\gamma}) = \mathbf{T}(\vec{\theta})$ because the probability of rankings restricted to each group are exactly the same.

To address this problem we consider an additional tensor decomposition using a different partition of the alternatives, defined as follows: $S'_1 = \{a_2, a_3\}, S'_2 = \{a_4, a_5\}, \ldots, S'_{m/2} = \{a_m, a_1\}$. Let $\mathbf{T}'(\vec{\theta})$ denote the tensor under this partition. Similar to the proof for Condition 1, we can show that generically $\mathbf{T}'(\vec{\theta})$ has a unique decomposition. Next, we will prove that for any $\vec{\theta}$ where $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$ have unique decompositions (which holds generically), Condition 2 holds.

Suppose for the sake of contradiction that Condition 2 does not hold for $\vec{\theta}$ where $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$ have unique decompositions. Then, there exists $\vec{\gamma} = (\vec{\beta}, \vec{\gamma}^{(1)}, \ldots, \vec{\gamma}^{(k)})$ that is different from $\vec{\theta}$ modulo label switching, such that $\Pr_{k\text{-RUM}_{\mathcal{M}}}(\cdot|\vec{\gamma}) = \Pr_{k\text{-RUM}_{\mathcal{M}}}(\cdot|\vec{\theta})$. This means that $\mathbf{T}(\vec{\gamma}) = \mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\gamma}) = \mathbf{T}'(\vec{\theta})$.

We first match the components in $\vec{\gamma}$ and $\vec{\theta}$. Recall that uniqueness of $\mathbf{T}(\vec{\theta})$ implies Condition 1. Because both $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$ have unique decompositions, Condition 1 must hold for both. It follows that the entries of $\vec{\beta}$ are exactly entries of $\vec{\alpha}$, otherwise $\mathbf{T}(\vec{\theta}) \neq \mathbf{T}(\vec{\gamma})$, which is a contradiction. Since entries of $\vec{\alpha}$ are pairwise different, there is a unique way of matching components in $\vec{\theta}$ to components in $\vec{\gamma}$ by matching the corresponding mixing coefficients. Consequently, we can relabel components in $\vec{\gamma}$ s.t. the $\vec{\beta}$ becomes exactly $\vec{\alpha}$. Let $\vec{\gamma}' = (\vec{\alpha}, \vec{\gamma}'^{(1)}, \ldots, \vec{\gamma}'^{(k)})$ denote the $\vec{\gamma}$ parameter after relabeling the components. Thus we have $\mathbf{T}(\vec{\gamma}') = \mathbf{T}(\vec{\gamma}) = \mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\gamma}') = \mathbf{T}'(\vec{\gamma}) = \mathbf{T}'(\vec{\theta})$. Because $\vec{\gamma} \neq \vec{\theta}$ modulo label switching, we have $\vec{\gamma}' \neq \vec{\theta}$, which implies that there exists r^* s.t.

$$\vec{\gamma}'^{(r^*)} \neq \vec{\theta}^{(r^*)}. \tag{3}$$

Next, we will show that from the uniqueness of decomposition of $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$, we must have $\vec{\gamma}'^{(r^*)} = \vec{\theta}^{(r^*)}$, which is a contradiction. To this end, we show how to uniquely determine the parameters of k-RUM$_{\mathcal{M}}$ (i.e. $\vec{\gamma}'$ and $\vec{\theta}$) from decompositions of $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$.

$\mathbf{T}(\vec{\gamma}') = \mathbf{T}(\vec{\theta})$ has a unique decomposition means that for any $r_1 \in \{1, \ldots, k\}$, there exists $r_2 \in \{1, \ldots, k\}$ s.t. $\alpha_{r_1}\mathbf{T}^{(r_1)}(\vec{\gamma}'^{(r_1)}) = \alpha_{r_2}\mathbf{T}^{(r_2)}(\vec{\theta}^{(r_2)})$, which implies the sums of all entries of $\alpha_{r_1}\mathbf{T}^{(r_1)}(\vec{\gamma}'^{(r_1)})$ and $\alpha_{r_2}\mathbf{T}^{(r_2)}(\vec{\theta}^{(r_2)})$ are equal, i.e. $\alpha_{r_1} = \alpha_{r_2}$. It follows from (1) that $r_1 = r_2$. Namely, we have $\alpha_r\mathbf{T}^{(r)}(\vec{\gamma}'^{(r)}) = \alpha_r\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$ for all $r = 1, \ldots, k$. It follows that for all $r = 1, \ldots, k$, we have $\mathbf{T}^{(r)}(\vec{\gamma}'^{(r)}) = \mathbf{T}^{(r)}(\vec{\theta}^{(r)})$, which means $\mathbf{T}^{(r^*)}(\vec{\gamma}'^{(r^*)}) = \mathbf{T}^{(r^*)}(\theta^{(r^*)})$. Similarly, we also have $\mathbf{T}'^{(r^*)}(\vec{\gamma}'^{(r^*)}) = \mathbf{T}'^{(r^*)}(\theta^{(r^*)})$.

For any r and q, $\mathbf{p}_q^{(r)}$ can be easily obtained from $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$ by normalizing the corresponding entries of $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$. E.g., $\mathbf{p}_1^{(r)}$ can be obtained by normalizing entries $(1, 1, \ldots, 1)$ and $(2, 1, \ldots, 1)$. Such $\mathbf{p}_1^{(r)}$ is uniquely determined by $\mathbf{T}^{(r)}(\vec{\theta}^{(r)})$ because the two entries of $\mathbf{p}_1^{(r)}$ sum up to 1. Specifically, for all $q = 1, \ldots, m/2$, $\mathbf{p}_q^{(r^*)}$ is uniquely determined by $\mathbf{T}^{(r^*)}(\vec{\theta}^{(r^*)})$ and $\mathbf{p}_q'^{(r^*)}$ is uniquely determined by $\mathbf{T}'^{(r^*)}(\vec{\theta}^{(r^*)})$.

Next, we will prove that for all $q = 1, \ldots, m/2$, $\mathbf{p}_q^{(r^*)}, \mathbf{p}_q'^{(r^*)}$ uniquely determine $\vec{\theta}^{(r^*)}$, which contradicts (3). Now we focus on $\vec{\gamma}'^{(r^*)}$ and $\vec{\theta}^{(r^*)}$ restricted to S_1. We claim that there exists a constant C_1 s.t. for all $i = 1, 2$ (i.e. $a_i \in S_1$), $\gamma_i'^{(r^*)} = \theta_i^{(r^*)} + C_1$. The reason is as follows. Recall from (2) that $\mathbf{p}_q^{(r^*)}$ consists of probabilities of $a_1 \succ a_2$ and $a_2 \succ a_1$ given the component r^*. By Proposition 1 in (Azari Soufiani, Parkes, and Xia 2014), $\Pr(a_1 \succ a_2|\vec{\theta}^{(r^*)})$ is a function of $\theta_1^{(r^*)} - \theta_2^{(r^*)}$ and $\Pr(a_1 \succ a_2|\vec{\gamma}'^{(r^*)})$ is a function of $\gamma_1'^{(r^*)} - \gamma_2'^{(r^*)}$. By matching the probabilities we have $\gamma_1'^{(r^*)} - \gamma_2'^{(r^*)} = \theta_1^{(r^*)} - \theta_2^{(r^*)}$, which means that there exists a constant C_1 s.t. for $i = 1, 2$, $\gamma_i'^{(r^*)} = \theta_i^{(r^*)} + C_1$.

Similarly for all $q = 1, \ldots, m/2$ there exist C_q, s.t.

$$\gamma_i'^{(r^*)} = \theta_i^{(r^*)} + C_q, \text{ for } i = 2q - 1, 2q \tag{4}$$

Similarly from $\mathbf{T}'^{(r)}(\vec{\gamma}^{(r)}) = \mathbf{T}'^{(r)}(\vec{\theta}^{(r)})$, for all $q = 1, \ldots, m/2$, there exists C'_q s.t.

$$\gamma_i'^{(r^*)} = \theta_i^{(r^*)} + C'_q, \text{ for } i = 2q, (2q + 1 \bmod m) \tag{5}$$

Therefore, we have $C_1 = C'_1$ by letting $i = 2$ in (4) and (5); $C'_1 = C_2$ by letting $i = 3$; $C_2 = C'_2$ by letting $i = 4$, etc. So we have $C_1 = \cdots = C_q = C'_1 = \cdots = C'_q$. Let $C_q = C'_q = C$ for all $q = 1, \ldots, m/2$. Then for all $i = 1, \ldots, m$, we have $\gamma_i'^{(r^*)} = \theta_i^{(r^*)} + C$. Because $\sum_{i=1}^m \gamma_i'^{(r^*)} = \sum_{i=1}^m \theta_i^{(r^*)} = 0$, we have $C = 0$, which contradicts (3).

As we have proved, the Lebesgue measure of parameters where tensor $\mathbf{T}(\vec{\theta})$ (or $\mathbf{T}'(\vec{\theta})$) has a nonunique decomposition is 0. Therefore, the Lebesgue measure of parameters $\vec{\theta}$ where both $\mathbf{T}(\vec{\theta})$ and $\mathbf{T}'(\vec{\theta})$ have unique decompositions is also 0. This finishes the proof. ■

The E-GMM Algorithm

To compute k-RUM$_{\mathcal{M}}$, we propose an EM-based algorithm, which we call E-GMM, where the GMM algorithm proposed by Azari Soufiani, Parkes, and Xia (2014) is used in the M step. The detailed algorithm is as follows.

During the E-step, given the t-th step estimate $\vec{\alpha}^{(t)}$ and $\vec{\theta}^{(1,t)}, \ldots, \vec{\theta}^{(k,t)}$, we use Bayes' rule to calculate the probability of a ranking V_j to belong to component r for $(t+1)$-th iteration, denoted as $p_{t+1}^{(jr)}$. We have

$$p_{t+1}^{(jr)} \propto \alpha_r^{(t)} \Pr(V_j|\vec{\theta}^{(r,t)}) \tag{6}$$

We can normalize w.r.t. r because $\sum_{r=1}^{k} p^{(jr)} = 1$.

In the M-step, mixing probabilities are calculated by

$$\alpha_r^{(t+1)} = \frac{\sum_{j=1}^{n} p_{t+1}^{(jr)}}{n} \tag{7}$$

and $\vec{\theta}^{(r,t+1)}$'s are estimated using the GMM Algorithm from (Azari Soufiani, Parkes, and Xia 2014). Formally, the E-GMM algorithm is presented below as Algorithm 1.

Algorithm 1 E-GMM Algorithm

```
Input: Profile $P$ of $n$ rankings, the number of components $k$, the number of iterations $T$.
Output: $\alpha_r^{(T+1)}$, $\vec{\theta}^{(r,T+1)}$, where $r = 1, 2, \cdots, k$.
Initialize $\alpha_r^{(1)}$, and $\vec{\theta}^{(r,1)}$ randomly for all $r = 1, 2, \cdots, k$.
for $t = 1$ to $T$ do
    Given the estimate at $t$-th step $\alpha_r^{(t)}$, $\vec{\theta}^{(r,t)}$.
    E step: calculate the expected membership by (6).
    M step: calculate $\alpha_r^{(t+1)}$ using (7) and use GMM with weighted rankings/breakings to estimate $\theta^{(r,t+1)}$.
end for
```

GMM for k-RUM$_{\mathcal{M}}$

Generalized-method-of-moments (GMM) (Hansen 1982) algorithms are a widely applied class of algorithms that generalize the classical method of moments. Each GMM is specified by a set of $q \geq 1$ *moment conditions* $g(V, \vec{\theta})$, where V is a data point and $\vec{\theta}$ is a parameter, such that for any $\vec{\theta}_0$, the expectation of each moment condition is zero at $\vec{\theta}_0$, when the data are generated from the model given $\vec{\theta}_0$. That is, $E[g(V, \vec{\theta}_0)] = \vec{0}$. Then, the algorithm computes $\vec{\theta}$ to minimize a certain norm, e.g. the 2-norm, of $\sum_{V \in P} g(V, \theta)$.

Our GMM algorithm for k-RUM$_{\mathcal{M}}$ is defined as follows. We first define a *breaking matrix* $B(P) = [b_{i_1 i_2}]_{m \times m}$, where $b_{i_1 i_2}$ is the empirical probability that a_{i_1} is preferred over a_{i_2}, namely the number of times that $a_{i_1} \succ a_{i_2}$ over the number of rankings. The diagonal elements are zeros. For example, let $P = (a_1 \succ a_2 \succ a_3, a_1 \succ a_3 \succ a_2)$, we have $B(P) = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix}$. We then define the following $\binom{m}{2}$ moment conditions: for any $i_1 < i_2$, let $g_{i_1 i_2}(P, \vec{\theta}) = b_{i_1 i_2} - \Pr(a_{i_1} \succ a_{i_2}|\vec{\theta})$. In our GMM algorithm, we minimize the following objective function

$$G = \sum_{i_2 > i_1} (b_{i_1 i_2} - \Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}))^2 \tag{8}$$

where $\Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}^{(r)})$. Formally, the algorithm is presented as Algorithm 2.

Algorithm 2 GMM for k-RUM$_{\mathcal{M}}$

```
Input: A Preference profile $P$.
Compute the breaking matrix $B(P)$.
Compute the parameter that minimizes (8).
```

The advantages of our GMM algorithm are:

1. Gradient and Hessian of G are easy to compute. To calculate the gradient of G, we need partial derivatives of $\Pr(a_{i_1} \succ a_{i_2}|\vec{\theta})$, which can be broken down to partial derivatives of $\Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}^{(r)})$ w.r.t. $\vec{\theta}^{(r)}$. This was given by Proposition 1 by (Azari Soufiani, Parkes, and Xia 2014).

2. $\Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}^{(r)})$ can be computed easily by sampling the utilities from the utility distributions of a_{i_1} and a_{i_2}. For Gaussian distributions, $\Pr(a_{i_1} \succ a_{i_2}|\vec{\theta}^{(r)})$ is the CDF valued at $\theta_{i_1}^{(r)} - \theta_{i_2}^{(r)}$ of another Gaussian distribution with mean 0 and variance being the sum of the variances of utility distributions of a_{i_1} and a_{i_2}.

For better accuracy we add m more moment conditions, corresponding to m cyclic triple-wise comparisons, i.e. $T_1 = a_1 \succ a_2 \succ a_3, T_2 = a_2 \succ a_3 \succ a_4, \ldots, T_m = a_m \succ a_1 \succ a_2$. Let b_{T_i} be the empirical probability of T_i. The new GMM minimizes: $G' = G + \sum_{i=1}^{m}(b_{T_i} - \Pr(T_i|\vec{\theta}))^2$.

We now prove that our GMM algorithm is *consistent*: when the data are generated independently from k-RUM$_{\mathcal{M}}$ and the size of data approaches infinity, the algorithm converges to the ground truth with probability that goes to 1.

Theorem 3 *If k-RUM$_{\mathcal{M}}$ over m alternatives is identifiable, and the means of the utility distributions of all alternatives in all RUM components are bounded in close intervals $[0, C]$, then Algorithm 2 is consistent.*

Proof: Hall (2005) provides a set of necessary conditions for any GMM to be consistent. Therefore, it suffices to check that all assumptions in Theorem 3.1 in (Hall 2005) holds.

Assumption 3.1: Strict Stationarity: the $(n \times 1)$ random vectors $\{v_t; -\infty < t < \infty\}$ form a strictly stationary process with sample space $\mathcal{S} \subseteq \mathbb{R}^n$. As the data are generated i.i.d., the process is strictly stationary.

Assumption 3.2: Regularity Conditions for $g(\cdot, \cdot)$: the function $g : \mathcal{S} \times \Theta \to \mathbb{R}^q$ where $q < \infty$, satisfies: (i) it is continuous on Θ for each $P \in \mathcal{S}$; (ii) $E[g(P, \vec{\theta})]$ exists and is finite for every $\theta \in \Theta$; (iii) $E[g(P, \vec{\theta})]$ is continuous on Θ. Our moment conditions satisfy all the regularity conditions since $g_{i_1 i_2}(P, \vec{\theta})$ is continuous on Θ and bounded in $[-1, 1]$ for any $i_2 \neq i_1$.

Assumption 3.3: Population Moment Condition. The random vector v_t and the parameter vector $\vec{\theta}_0$ satisfy the $(q \times 1)$

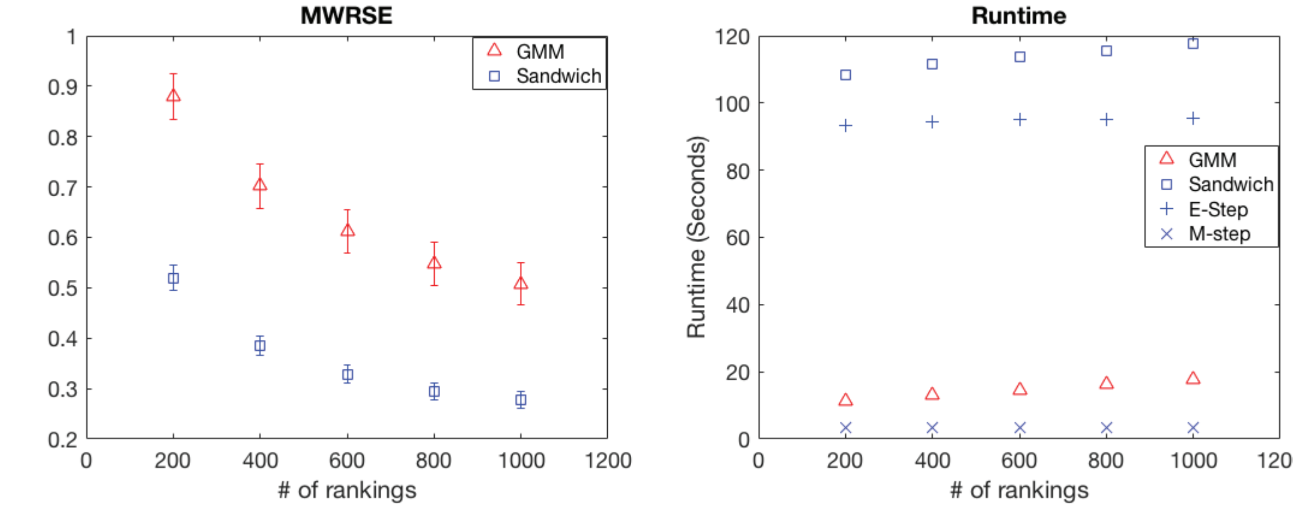

Figure 1: MWRSE with 95% confidence intervals and runtime for 2-RUM over 6 alternatives. We use 10 EM iterations for the sandwich algorithm. Values are averaged over 1000 trials. The ground truth for each component ranges between 0 and 5.

population moment condition: $E[g(P, \vec{\theta}_0)] = 0$. This assumption holds by the definition of our GMM.

Assumption 3.4: Global Identification. $E[g(P, \vec{\theta'})] \neq 0$ for all $\vec{\theta'} \in \Theta$ such that $\vec{\theta'} \neq \theta_0$. This is the assumption of the theorem.

Assumption 3.7: Properties of the Weighting Matrix. W_t is a positive semi-definite matrix which converges in probability to the positive definite matrix of constants W. This holds because $W = I$.

Assumption 3.8: Ergodicity. The random process $\{v_t; -\infty < t < \infty\}$ is ergodic. Since the data are generated i.i.d., the process is ergodic.

Assumption 3.9: Θ is a compact set. The mixing probabilities are compact in interval $[0, 1]$ and in the assumption of this theorem, the $\theta_i^{(r)}$s are bounded in $[0, C]$.

Assumption 3.10: Domination of $g(P, \vec{\theta})$. $E[\sup_{\theta\in\Theta} ||g(P, \vec{\theta})||] < \infty$. This assumption holds because all moment conditions are finite. ■

The Sandwich Algorithm

The performance of E-GMM algorithm is sensitive to the initial value. Therefore, we propose the sandwich algorithm (GMM-E-GMM) to give our E-GMM algorithm a good starting point. We note that the first GMM stands for our GMM algorithm for learning k-RUMs and the second GMM stands for Azari Soufiani, Parkes, and Xia's algorithm [2014] for learning a single RUM in each M-step. The algorithm is formally shown as Algorithm 3.

Algorithm 3 Sandwich (GMM-E-GMM) Algorithm

Input: Profile P of n rankings, the number of components k, the number of iterations T.
Output: α_r, $\vec{\theta}^{(r)}$, where $r = 1, 2, \cdots, k$.

```
Run Algorithm 2, whose output is $\alpha_r^{(0)}$, $\vec{\theta}^{(r,0)}$, where $r = 1, 2, \cdots, k$.
Use the output as the initial value of Algorithm 1.
```

Experiments

We implemented all algorithms with Matlab and tested them on synthetic data and Preflib data.

Synthetic data generation. In each trial, we first generate the ground truth parameters for k RUM components and normalize each $\vec{\theta}^{(r)}$ s.t. $\sum_{i=1}^{m} \theta_i^{(r)} = 0$. The mixing coefficients $\vec{\alpha}$ are generated uniformly at random in $[0, 1]$ and then normalized s.t. $\sum_{r=1}^{k} \alpha_r = 1$. Then with probability α_r, the rth component is selected to generate a full ranking. We run experiments on two settings: (i) $k = 2, m = 6$, 1000 trials (Figure 1), and (ii) $k = 4, m = 15$, 900 trials (Figure 2). In both figures, results for the E-GMM algorithm are not shown because the sandwich algorithm strictly improves E-GMM w.r.t. both statistical efficiency and computational efficiency. All experiments were run on an Ubuntu Linux server with Intel Xeon E5 v3 CPUs clocked at 3.50 GHz.

Measures. We note that components with small mixing coefficients are generally hard to learn accurately—in the extreme case, when the mixing coefficient for one component is 0, it is impossible to learn the component. Therefore, the standard mean square error is not very informative for our experiments. Consequently, we define MWRSE (mean weighted root square error) to mitigate the impact of such components. Formally, let $\vec{\theta}_0$ denote the ground truth parameter and $\vec{\theta'}$ denote the estimate. We define WRSE as follows:

$$\text{WRSE} = ||\vec{\alpha}_0 - \vec{\alpha}'||_2 + \sum_{r=1}^{k} \alpha_{0,r} ||\vec{\theta}_0^{(r)} - \vec{\theta}'^{(r)}||_2$$

MWRSE is computed by averaging WRSE of all trials.

Observations. The sandwich algorithm has lower MWRSEs and narrower 95% confidence than GMM under both set-

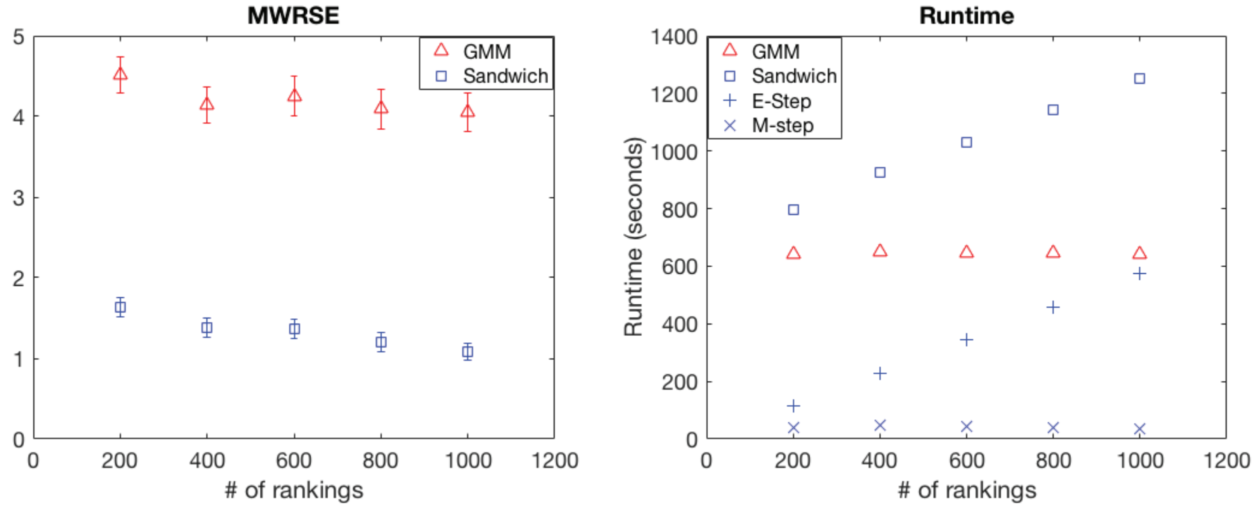

Figure 2: MWRSE with 95% confidence intervals and runtime for 4-RUM over 15 alternatives. We use 5 EM iterations for the sandwich algorithm. Values are averaged over 900 trials. The ground truth for each component ranges between 0 and 10.

tings (left subfigures of Figures 1 and 2), which means the sandwich algorithm achieves better statistical efficiency. In terms of runtime, the sandwich algorithm is not as fast as GMM (right subfigures of Figures 1 and 2). Moreover, the running time of the E step dominates that of the M step.

Our sandwich algorithm achieves satisfactory statistical efficiency based on the following high-level reasoning. From Figure 1 we observe that the MWRSE for the sandwich algorithm is no more than 0.6 for $k = 2, m = 6$. Therefore, typically the root squared error for each component is about 0.6 because the weights sum up to 1. For each single parameter, we would expect the error to be typically $\sqrt{0.6^2/6} \approx 0.245$, which is reasonably small considering the range of each single parameter to be $[0, 5]$. Similarly for $k = 4, m = 15$, the error of each single parameter is typically below $\sqrt{2^2/15} \approx 0.52$, which is also small compared to the range $[0, 10]$.

Real-World Data. We learn different models, including the Plackett-Luce model (PL), its mixtures (k-PLs), and Gaussian k-RUMs from 209 linear order datasets on Preflib (Mattei and Walsh 2013) and compute their AIC, AICc and BIC, defined as: AIC $= 2d - 2\ln(L)$, AICc $=$ AIC$+ \frac{2d(d+1)}{n-d-1}$ and BIC $= d\ln(n) - 2\ln(L)$, where L is the value of the likelihood function evaluated at the estimation, d is the number of parameters in the model, and n is the number of rankings. A smaller AIC, AICc or BIC means better fitness.

For mixture models (k-PL and k-RUM), we increase k until all the three measurements start increasing. We compute the percentage for one model to be strictly better (lower) than another w.r.t. each measure, shown in Table 1. For example, in terms of AIC, k-RUM (with the best k) beats a single RUM in 60.3% of the datasets, which means that in 60.3% of the datasets, the best k for k-RUM is at least 2.

Observations. From Table 1 we can see that the three information criteria agree on the following order of models:

$$k\text{-RUM} \succ k\text{-PL} \succ \text{RUM} \succ \text{PL},$$

where $A \succ B$ means that the number of datasets where A beats B is more than that where B beats A.

		k-RUM	k-PL	RUM	PL
AIC	k-RUM	0	60.8%	60.3%	90.0%
	k-PL	39.2%	0	79.4%	90.4%
	RUM	0	20.6%	0	76.6%
	PL	10.0%	0	23.4%	0
AICc	k-RUM	0	60.3%	59.8%	90.0%
	k-PL	39.7%	0	79.4%	89.5%
	RUM	0	20.6%	0	76.6%
	PL	10.0%	0	23.4%	0
BIC	k-RUM	0	66.0%	40.2%	84.2%
	k-PL	34.0%	0	59.8%	66.0%
	RUM	0	40.2%	0	76.6%
	PL	15.8%	0	23.4%	0

Table 1: Model fitness comparisons.

Conclusions and Future Work

We characterize conditions for mixtures of general RUMs to be non-identifiable and generically identifiable, and designed three algorithms for computing them. Our experiments show that the sandwich algorithm achieves higher statistical efficiency and GMM achieves higher computational efficiency. Open questions include improving the identifiability theorems for k-RUMs and designing more efficient algorithms, such as tensor-decomposition-based algorithms.

Acknowledgments

We thank all anonymous reviewers for helpful comments and suggestions. This work is supported by NSF #1453542 and ONR #N00014-17-1-2621.

References

Akaike, H. 1974. A new look at the statistical model identification. *IEEE transactions on automatic control* 19(6):716–723.

Anandkumar, A.; Ge, R.; Hsu, D.; Kakade, S. M.; and Telgarsky, M. 2014. Tensor decompositions for learning latent variable models. In 15., ed., *The Journal of Machine Learning Research*, volume 1, 2773–2832.

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

Azari Soufiani, H.; Parkes, D. C.; and Xia, L. 2012. Random utility theory for social choice. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 126–134.

Azari Soufiani, H.; Parkes, D. C.; and Xia, L. 2014. Computing parametric ranking models via rank-breaking. In *Proceedings of the 31st International Conference on Machine Learning*.

Chen, Y., and Suh, C. 2015. Spectral MLE: Top-k rank aggregation from pairwise comparisons. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37.

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)* 1–38.

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

Hall, A. R. 2005. *Generalized Method of Moments*. Oxford University Press.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50(4):1029–1054.

Hurvich, C. M., and Tsai, C.-L. 1989. Regression and time series model selection in small samples. *Biometrika* 76(2):297–307.

Khetan, A., and Oh, S. 2016a. Computational and statistical tradeoffs in learning to rank. In *Advances in Neural Information Processing Systems (NIPS)*.

Khetan, A., and Oh, S. 2016b. Data-driven rank breaking for efficient rank aggregation. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48.

Liu, T.-Y. 2011. *Learning to Rank for Information Retrieval*. Springer.

Luce, R. D. 1959. *Individual Choice Behavior: A Theoretical Analysis*. Wiley.

Mattei, N., and Walsh, T. 2013. Preflib: A library of preference data. In *Proceedings of Third International Conference on Algorithmic Decision Theory (ADT 2013)*, Lecture Notes in Artificial Intelligence.

Mollica, C., and Tardella, L. 2016. Bayesian Plackett–Luce mixture models for partially ranked data. *Psychometrika* 1–17.

Negahban, S.; Oh, S.; and Shah, D. 2012. Iterative ranking from pair-wise comparisons. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2483–2491.

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

Redner, R. A., and Walker, H. F. 1984. Mixture densities, maximum likelihood and the EM algorithm. *SIAM review* 26(2):195–239.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6(2):461–464.

Sidiropoulos, N. D., and Bro, R. 2000. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of chemometrics* 14(3):229–239.

Thurstone, L. L. 1927. A law of comparative judgement. *Psychological Review* 34(4):273–286.

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. ACM.

Zhao, Z.; Piech, P.; and Xia, L. 2016. Learning mixtures of Plackett-Luce models. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*.