

Learning Plackett-Luce Mixtures from Partial Preferences

Ao Liu

Rensselaer Polytechnic Institute
Troy, NY 12180, USA
liua6@rpi.edu

Zhibing Zhao

Rensselaer Polytechnic Institute
Troy, NY 12180, USA
zhaoz6@rpi.edu

Chao Liao

Shanghai Jiao Tong University
Shanghai 200240, China
chao.liao.95@gmail.com

Pinyan Lu

Shanghai University of Finance and Economics
Shanghai 200433, China
lu.pinyan@mail.shufe.edu.cn

Lirong Xia

Rensselaer Polytechnic Institute
Troy, NY 12180, USA
xial@cs.rpi.edu

Abstract

We propose an EM-based framework for learning Plackett-Luce model and its mixtures from *partial orders*. The core of our framework is the efficient sampling of linear extensions of partial orders under Plackett-Luce model. We propose two Markov Chain Monte Carlo (MCMC) samplers: Gibbs sampler and the generalized repeated insertion method tuned by MCMC (GRIM-MCMC), and prove the efficiency of GRIM-MCMC for a large class of preferences.

Experiments on synthetic data show that the algorithm with Gibbs sampler outperforms that with GRIM-MCMC. Experiments on real-world data show that the likelihood of test dataset increases when (i) partial orders provide more information; or (ii) the number of components in mixtures of Plackett-Luce model increases.

Introduction

Rank aggregation has found applications in many areas, such as meta-search (Dwork et al. 2001), information retrieval (Liu 2011), and recommender systems (Baltrunas, Makcinskis, and Ricci 2010). In the rank aggregation problem, we want to find an aggregated ranking from the rank data of a given population. One popular approach is *learning to rank*, which is to learn a statistical ranking model from the data, and interpret the learned parameter as a ranking.

Plackett-Luce model (PL) (Plackett 1975; Luce 1959) is one of the most widely-used models. In PL, each alternative is assigned a positive value. The greater this value is, the more likely its corresponding alternative is ranked at a higher position. In recent years, mixtures of Plackett-Luce models have drawn attention for heterogeneous data. Mixture models provide better fitness to data (Zhao, Villamil, and Xia 2018), and can be used for clustering. A mixture model assumes that there are multiple clusters in the population, and each of the clusters can be learned using a component model. In the case of mixtures of Plackett-Luce models, each component is a Plackett-Luce model, while the quality of a certain alternative is usually different across different components.

Algorithms to learn a single Plackett-Luce model and its mixtures from *linear orders* (full rankings) have been well investigated. However, many real-world datasets are composed of *partial orders* over a subset of alternatives (Lu and

Boutillier 2014; Hüllermeier et al. 2008), e.g. the Netflix prize data. Some of the previously proposed algorithms work only for special structures of partial orders or general partial orders with ignored comparisons or dependence between pairwise comparisons. However, modern datasets can be unstructured or even implicit, e.g., preferences implied by comparing time users spent on news articles or whether a user finished watching a movie (Khetan and Oh 2016b). None of the existing algorithms is able to learn PL and its mixtures with full utilization of general partial orders with no restrictions. The only known approach to tackle general partial orders without ignoring information in literature is to sample linear extensions from partial orders under a given model (Mallows' model) and learn the model from full rankings (Lu and Boutillier 2014).

This paper addresses the following problem: *How can we efficiently learn the Plackett-Luce model and its mixtures from partial orders?*

Our Contributions. We propose an EM framework to learn mixtures of Plackett-Luce models from partial orders. The E step consists of two parts: given the current estimate, (i) sampling multiple linear extensions from each partial order, and (ii) computing the membership of each sampled linear order (full ranking). The M step can adapt any learning algorithm for the single Plackett-Luce model. In this paper, we use the LSR algorithm by Maystre and Grossglauser (2015) as the subroutine.

We address the technical challenge of sampling a linear extension of a partial order under PL in part (i) by proposing the following two samplers.

Sampler 1: GRIM-MCMC generalizes the GRIM-based sampling algorithm by Lu and Boutillier (2014), which was for Mallows' model, to PL. We further prove that GRIM-MCMC is efficient for a large class of preferences called *layered-structure preferences* in Theorem 4.

Sampler 2: Gibbs generalizes the sampler by Azari Soufiani, Parkes, and Xia (2012) to deal with partial orders, and can be naturally generalized to learn mixtures of Random Utility Models (Thurstone 1927), which subsume PL.

We compare the performances of the proposed sampling algorithms for PL and its mixtures using experiments on synthetic data and show that Gibbs sampler is better. Our algorithm uses 75% information at the cost of mere 6% MSE compared to the best algorithm for PL mixtures for linear or-

ders. Experiments on real-world data show that the likelihood of test dataset increases when partial orders provide more information; or the number of components in PL mixtures increases within a certain range.

Related Work. We are not aware of any previous algorithm that learns a mixture of Plackett-Luce models from general partial orders without discarding information. A vast number of algorithms have been proposed to learn the Plackett-Luce model (Hunter 2004; Azari Soufiani et al. 2013; Maystre and Grossglauser 2015; Khetan and Oh 2016b; 2016a; Negahban, Oh, and Shah 2017; Zhao and Xia 2018) and its mixtures (Gormley and Murphy 2008; Oh and Shah 2014; Mollica and Tardella 2016; Tkachenko and Lauw 2016; Zhao, Piech, and Xia 2016). Some of these algorithms were designed for linear orders (full rankings) (Azari Soufiani et al. 2013; Zhao and Xia 2018; Liu et al. 2019; Zhao, Piech, and Xia 2016), or some special structures of partial orders (Hunter 2004; Maystre and Grossglauser 2015; Khetan and Oh 2016b; Negahban, Oh, and Shah 2017; Mollica and Tardella 2016; Tkachenko and Lauw 2016; Zhao et al. 2018), e.g. top- l rankings. Very few algorithms were designed for general partial orders, but some ignore dependencies between pairwise comparisons with overlapping alternatives (Oh and Shah 2014), and some discard comparison information in order for the algorithm to be consistent (Khetan and Oh 2016a).

Lu and Boutilier (2014) proposed an algorithm to learn the Mallows model and its mixtures from general partial orders by sampling linear extensions. We focus on a very different but even more popular model: Plackett-Luce model, as well as its mixtures. Efficiently sampling linear extensions from arbitrary distributions is an open problem. There have been research on generating linear extensions according to various distributions (e.g., uniform distribution, Mallows model, etc.). Repeated insertion methods (RIM) and MCMC are two groups of methods widely used in those works.

We propose two algorithms to sample linear extensions from a partial order under the Plackett-Luce model: one inspired by Lu and Boutilier (2014) while the other based on the random utility nature of Plackett-Luce model. To our best knowledge, this is the first work that exploits all information from general partial orders by sampling linear extensions under the Plackett-Luce model, as well as other random utility models (Gibbs).

Preliminaries

Let $\mathcal{C} = \{c_1, \dots, c_m\}$ denote a set of m alternatives. Let $\mathcal{L}(\mathcal{C})$ denote the set of linear orders, which are transitive, antisymmetric and total binary relations, over \mathcal{C} . A linear order R is often denoted by $c_{i_1} \succ c_{i_2} \succ \dots \succ c_{i_m}$, which means that c_{i_1} is ranked at the top, c_{i_2} is ranked at the second, ..., c_{i_m} is ranked at the bottom. Let $\{1, \dots, n\}$ denote n agents. Each agent j is assumed to have a strict complete preference over \mathcal{C} , represented by a linear order R_j . However, only partial preferences over a subset of alternatives \mathcal{ES}_j is observed, represented by a *partial order*¹, which consists of irreflexive, transitive, and asymmetric binary relations. We call \mathcal{ES}_j the

¹Throughout this paper, all partial orders are strict.

effective alternative set of agent j and let $m_j^* = |\mathcal{ES}_j|$. We describe a partial order by a set of pairwise comparisons $V_j = \{c_{i_1} \succ c_{i'_1}, \dots, c_{i_q} \succ c_{i'_q} | 1 \leq i_1, i'_1, \dots, i_q, i'_q \leq m_j^*\}$. Due to transitivity, such description of a partial order is not unique. A unique representation of a partial order V_j is the transitive closure, denoted by $\text{tc}(V_j)$, which consists of all pairwise comparisons in V_j considering transitivity. An example of a partial order and its transitive closure is shown in Figure 1. Let $P = (V_1, \dots, V_n)$ denote the data (also called a *preference profile*), where V_j is agent j 's partial preference. Let $\mathcal{P}(\mathcal{C})$ denote the set of all partial orders.

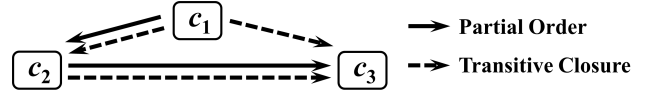


Figure 1: Example of a partial order and its transitive closure.

Let $\text{Ext}(V_j)$ be the set of all *linear extensions* of V_j , which consists of all linear orders consistent with V_j . A linear extension bridges a partial order (observed data) and a linear order. For example, $\text{Ext}(\emptyset)$ is the set of all permutations on \mathcal{C} . In this paper, we assume each agent's preference is generated from the Plackett-Luce model, defined as follows.

Definition 1 (Plackett-Luce model) *The sample space is $\mathcal{L}(\mathcal{C})^n$. The parameter space is $\Theta = \{\vec{\theta} = (\theta_1, \dots, \theta_m) \mid \forall i, \theta_i \in (0, 1), \sum_{i=1}^m \theta_i = 1\}$. Given parameter $\vec{\theta} \in \Theta$, the probability of any linear order $R = c_{i_1} \succ, \dots, \succ c_{i_m}$ is*

$$\text{Pr}_{\text{PL}}(R|\vec{\theta}) = \frac{\theta_{i_1}}{\sum_{p \geq 1} \theta_{i_p}} \times \frac{\theta_{i_2}}{\sum_{p \geq 2} \theta_{i_p}} \times \dots \times \frac{\theta_{i_{m-1}}}{\theta_{i_{m-1}} + \theta_{i_m}}. \quad (1)$$

The probability of any partial order V is the sum of all probability of its linear extensions, i.e. $\text{Pr}_{\text{PL}}(V|\vec{\theta}) = \sum_{R \in \text{Ext}(V)} \text{Pr}(R|\vec{\theta})$. Another probability of interest is the probability of linear order R conditioned on a partial order V , where $R \in \text{Ext}(V)$. By the definition of conditional probability, when $R \in \text{Ext}(V)$, we have

$$\text{Pr}_{\text{PL}}(R|\vec{\theta}, V) = \frac{\text{Pr}_{\text{PL}}(R|\vec{\theta})}{\text{Pr}_{\text{PL}}(V|\vec{\theta})}. \quad (2)$$

The Plackett-Luce model has the random utility interpretation, where each alternative c_i is associated with a Gumbel distribution centered at $\ln \theta_i$, whose cumulative distribution function is $e^{-\theta_i \cdot e^{-x}}$ (Yellott 1977). A linear order is generated by sampling a utility for each alternative from its corresponding utility distribution and ranking all the sampled utilities in descending order. For any $\vec{\theta} \in \Theta$, we further define $\eta_{\vec{\theta}} = \frac{\theta_{\max}}{\theta_{\min}}$, where $\theta_{\max} = \max_{i \in [m]} \theta_i$ and $\theta_{\min} = \min_{i \in [m]} \theta_i$.

Following (Ford Jr 1957) and (Hunter 2004), we make the following assumption on the data so that there exists a maximum likelihood estimation.

Assumption 1 *In every possible partition of the alternatives into two nonempty subsets, some individual in the second set beats some individual in the first set at least once.*

We define the mixture of k Plackett-Luce models (k -PL), which subsumes the Plackett-Luce model by letting $k = 1$, as follows.

Definition 2 (k -PL) Given integers $m \geq 2$ and $k \geq 1$, the k -mixture Plackett-Luce model is defined as follows. The sample space is $\mathcal{L}(C)^n$. The parameter space has two parts. The first part is the mixing coefficients $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$, where for all $1 \leq r \leq k$, $\alpha_r \geq 0$ and $\sum_{r=1}^k \alpha_r = 1$. The second part is $(\vec{\theta}^{(1)}, \dots, \vec{\theta}^{(k)})$, where $\vec{\theta}^{(r)} \in \Theta$ is the parameter of the r -th Plackett-Luce component. The probability of any linear order R is $\Pr_{k\text{-PL}}(R|\vec{\theta}) = \sum_{r=1}^k \alpha_r \Pr_{\text{PL}}(R|\vec{\theta}^{(r)})$.

Similar to the (single) Plackett-Luce model, the probability of any partial order V is $\Pr_{k\text{-PL}}(V|\vec{\theta}) = \sum_{R \in \text{Ext}(V)} \Pr_{k\text{-PL}}(R|\vec{\theta})$, which is equivalent to $\Pr_{k\text{-PL}}(V|\vec{\theta}) = \sum_{r=1}^k \alpha_r \Pr_{\text{PL}}(V|\vec{\theta}^{(r)})$.

An EM Framework for Learning PL Mixtures

We propose an EM framework where the E-step has two parts: (1) generating N linear extensions from each partial order, and (2) computing the expected membership of each linear order. In the next section, we will introduce the two sampling algorithms for part (1): GRIM-MCMC (Algorithm 3) and Gibbs Sampling (Algorithm 4).

Algorithm 1: EM Algorithm for k -PL with GRIM sampler and Gibbs sampler

- 1 **Input:** Profile $P = (V_1, \dots, V_n)$, the number of components k , number of linear extensions per partial order N , the number of iterations T .
 - 2 **Output:** For all $1 \leq r \leq k$, $\alpha_r^{(T)}$ and $\theta^{(r,T)}$
 - 3 **Initialization:** Randomly generate $\vec{\theta}^{(1,0)}, \dots, \vec{\theta}^{(k,0)}$, and mixing coefficients $\alpha_1^{(0)}, \dots, \alpha_k^{(0)}$.
 - 4 **for** $t = 1, \dots, T$ **do**
 - 5 E-step:
 - 6 Sample N linear extensions for each partial order using e.g. Algorithm 3 (GRIM-MCMC) or Algorithm 4 (Gibbs);
 - 7 $\forall 1 \leq j \leq n, 1 \leq r \leq k$, compute $w_{jr}^{(t)}$ using (3).
 - 8 M-step:
 - 9 For all $1 \leq r \leq k$, update $\alpha_r^{(t)} = \frac{\sum_j w_{jr}^{(t)}}{n}$ and compute $\vec{\theta}^{(r)}$ using the LSR algorithm by (Maystre and Grossglauser 2015).
 - 10 **end**
-

Formally, let z_{jr} denote the membership indicator where $z_{jr} = 1$ means the linear order R_j belongs to the r th component. Let w_{jr} denote the weight of ranking R_j in r th component. For all j , we have $\sum_r w_{jr} = 1$. Given last iteration estimate $(\vec{\alpha}^{(t-1)}, \vec{\theta}^{(1,t-1)}, \dots, \vec{\theta}^{(k,t-1)})$, each linear order

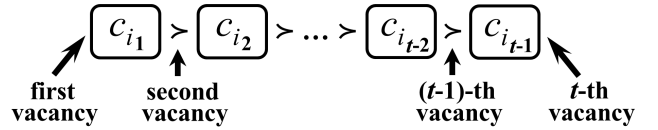


Figure 2: Definition and labeling of vacancies.

R_j is clustered to each component by the following equation

$$\begin{aligned} w_{jr}^{(t)} &= \Pr(z_{jr} = 1 | R_j, \vec{\theta}^{(r,t-1)}) \\ &= \frac{\Pr(R_j | \vec{\theta}^{(r,t-1)}) \cdot \alpha_r^{(t-1)}}{\sum_{s=1}^k \Pr(R_j | \vec{\theta}^{(s,t-1)}) \cdot \alpha_s^{(t-1)}}. \end{aligned} \quad (3)$$

In the case of $k = 1$, the membership computation is redundant.

In the M-step, we update the mixing coefficients by $\alpha_r^{(t)} = \frac{\sum_j w_{jr}^{(t)}}{n}$ and the parameters of all components $(\vec{\theta}^{(1)}, \dots, \vec{\theta}^{(k)})$ using the LSR algorithm proposed by (Maystre and Grossglauser 2015). The algorithm is formally shown as Algorithm 1.

Sampling Linear Extensions according to PL

In the section, we introduce two efficient algorithms to sample linear extensions of a partial order under PL, which are used in step 6 of Algorithm 1.

The GRIM-MCMC Sampler for Plackett-Luce

We extend the GRIM-based algorithm by (Lu and Boutilier 2014) to Plackett-Luce model. Repeated insertion model (RIM) (Doignon, Pekeč, and Regenwetter 2004) is a polynomial-time tool applicable to exactly sample from Plackett-Luce model (given an empty partial order). Then, we generalize the Plackett-Luce RIM to an approximate sampler for linear extensions given an arbitrary partial order.

GRIM for Plackett-Luce Model We first present the PL RIM sampler as the building block of GRIM. RIM samples a linear order by inserting alternatives one by one to the temporary ranking until the ranking is complete, where the order of insertions is arbitrary. W.l.o.g., we use the order from c_1 to c_m throughout this section. Let $R^{(t-1)} = [c_{r_1}^{(t-1)} \succ \dots \succ c_{r_{t-1}}^{(t-1)}]$ denote the temporary ranking over alternatives c_1, \dots, c_{t-1} . Let the i -th vacancy of this ranking as the i -th possible position for c_t to be inserted (shown in Figure 2), we define *conditional insertion probabilities* for PL RIM as:

$$\begin{aligned} &\Pr_{\text{RIM}} \left(R^{(t)} \mid R^{(t-1)} \right) \\ &\equiv \mathbf{1}_{R^{(t)} \in \text{Ext}(R^{(t-1)})} \cdot \frac{\Pr_{\text{PL}} \left(R^{(t)} \mid (\theta_1, \dots, \theta_t) \right)}{\Pr_{\text{PL}} \left(R^{(t-1)} \mid (\theta_1, \dots, \theta_{t-1}) \right)}. \end{aligned} \quad (4)$$

The condition in indicator function $\mathbf{1}_{R^{(t)} \in \text{Ext}(R^{(t-1)})}$ guarantees c_t is inserted to one of the vacancies of $R^{(t-1)}$. Due to the space limit, we only present the generalized version

of RIM in Algorithm 2, which will reduce to RIM if letting input V be \emptyset .

Theorem 1 *PL RIM is an exact sampler for Plackett-Luce model that runs in $O(m^2)$ time.*

Proof: We first prove the correctness of PL RIM. Slightly abusing the notation, we let $R^{(t)}$ be the temporary ranking over c_1, \dots, c_t following the same order as full ranking R . It's easy to check that $R^{(t)} \in \text{Ext}(R^{(t-1)})$ and $R = R^{(m)}$. Thus, for Plackett-Luce RIM defined above, we have, $\Pr_{\text{RIM}}(R|\vec{\theta}) = \Pr_{\text{RIM}}(R^{(m)}|R^{(m-1)}) \times \dots \times \Pr_{\text{RIM}}(R^{(2)}|R^{(1)}) \times \Pr_{\text{RIM}}(R^{(1)}) = \Pr_{\text{PL}}(R|\vec{\theta})$.

We now analyze the complexity of calculating all insertion probabilities at the t -th insertion step. In stead of directly calculating (4) from (1), we use the following iterative formula to calculate $\Pr_{\text{RIM}}(R_i^{(t)}|R^{(t-1)})$ from $\Pr_{\text{RIM}}(R_{i-1}^{(t)}|R^{(t-1)})$:

$$\frac{\Pr_{\text{RIM}}(R_i^{(t)}|R^{(t-1)})}{\Pr_{\text{RIM}}(R_{i-1}^{(t)}|R^{(t-1)})} = \frac{\sum_{i'=i-1}^{t-1} \theta_{r_{i'}^{(t-1)}}}{\theta_t + \sum_{i'=i}^{t-1} \theta_{r_{i'}^{(t-1)}}} = \frac{A_i^{(t-1)}}{B_i^{(t-1)}}.$$

It is noteworthy that $A_i^{(t-1)}$ and $B_i^{(t-1)}$ can be calculated through the iterative formulas $A_i^{(t-1)} = A_{i+1}^{(t-1)} + \theta_{r_{i+1}^{(t-1)}}$ and $B_i = A_{i+1}^{(t-1)} + \theta_t$. The 2-level iteration process gives a $O(t)$ approach to calculate all needed probabilities and Theorem 1 follows by combining all m insertion steps. \square

Algorithm 2: Plackett-Luce GRIM

- 1 **Input:** Partial order V , Plackett-Luce parameters $\vec{\theta} = (\theta_1, \dots, \theta_m)$ (θ_i corresponds to alternative c_i).
 - 2 **Initialization:** Set $R^{(1)} = [c_1]$ as initialized ranking.
 - 3 **for** $t = 2; t \leq m$ **do**
 - 4 Insert alternative c_t to $R^{(t-1)} = [c_{r_1}^{(t-1)} \succ c_{r_2}^{(t-1)} \succ \dots \succ c_{r_{t-1}}^{(t-1)}]$'s i^{th} vacancy ($1 \leq i \leq t$) with probability calculated by (5).
 - 5 **end**
 - 6 **return** $R^{(m)}$.
-

Now we are ready to propose PL GRIM, which samples linear extensions of any given partial order, where in each step we only insert the current alternative to vacancies consistent with the partial order. We define the GRIM insertion probability by

$$\Pr_{\text{GRIM}}(R_i^{(t)}|R^{(t-1)}) \equiv \frac{\mathbf{1}_{R_i^{(t)} \in \text{Ext}(V)}}{Z_{V, R^{(t-1)}, \vec{\theta}}} \cdot \frac{\Pr_{\text{PL}}(R_i^{(t)} | (\theta_1, \dots, \theta_t))}{\Pr_{\text{PL}}(R^{(t-1)} | (\theta_1, \dots, \theta_{t-1}))}, \quad (5)$$

where

$$Z_{V, R^{(t-1)}, \vec{\theta}} = \sum_{i=1}^t \left(\mathbf{1}_{R_i^{(t)} \in \text{Ext}(V)} \cdot \frac{\Pr_{\text{PL}}(R_i^{(t)} | (\theta_1, \dots, \theta_t))}{\Pr_{\text{PL}}(R^{(t-1)} | (\theta_1, \dots, \theta_{t-1}))} \right)$$

is the normalization factor. Formally, GRIM procedure is shown in Algorithm 2 (note again its fourth-step used (5) and thus used the information from V).

Theorem 2 *PL GRIM (Algorithm 2) runs in $O(m^2)$ time, and the probability of outputting full ranking R is $\Pr_{\text{GRIM}}(R|\vec{\theta}, V) = \frac{\mathbf{1}_{R \in \text{Ext}(V)}}{\prod_{t=1}^{m-1} Z_{V, R^{(t)}, \vec{\theta}}} \cdot \Pr_{\text{PL}}(R|\vec{\theta})$.*

Table 1: Example of GRIM (probability columns show conditional probabilities on given rankings)

Insert c_1, c_2		Insert c_3	
Ranking	Pr	Ranking	Pr
$c_1 \succ c_2$	$\frac{1}{1+2} = \frac{1}{3}$	$c_1 \succ c_3 \succ c_2$	3/5
		$c_1 \succ c_2 \succ c_3$	2/5
$c_2 \succ c_1$	$\frac{2}{1+2} = \frac{2}{3}$	$c_2 \succ c_1 \succ c_3$	1

The distribution of PL GRIM is not always exactly the distribution in (2) (See Example 1).

Example 1 *Consider the Plackett-Luce model over $\mathcal{C} = \{c_1, c_2, c_3\}$, partial order $V = \{[c_1 \succ c_3]\}$ and parameter $\vec{\theta} = (1/6, 2/6, 3/6)$, the process of GRIM is shown in Table 1. The comparison between GRIM and Plackett-Luce model is shown in Table 2.*

Table 2: Comparing the Distributions between GRIM PL

	$c_1 \succ c_3 \succ c_2$	$c_1 \succ c_2 \succ c_3$	$c_2 \succ c_1 \succ c_3$
\Pr_{GRIM}	1/5	2/15	2/3
\Pr_{PL}	2/5	4/15	1/3

Even though GRIM is not a precise sampler for Plackett-Luce model, it provides us a good proposal distribution on linear extensions for Metropolis–Hastings algorithm (Metropolis et al. 1953; Hastings 1970), which is an MCMC algorithm that tunes a proposal distribution to the target distribution. It works as follows. For each state β we first generate a candidate next state γ from a proposal distribution $\hat{\mathcal{P}}(\cdot)$ (slightly abusing this notation). Then, let the next state to be γ with probability $\min \left\{ 1, \frac{\mathcal{P}(\gamma)\hat{\mathcal{P}}(\beta|\gamma)}{\mathcal{P}(\beta)\hat{\mathcal{P}}(\gamma|\beta)} \right\}$, otherwise the next state remains β , where $\mathcal{P}(\cdot)$ is the (probability of) target distribution.

GRIM-MCMC In this section, we present the Markov Chain \mathcal{M}_{PL} used to tune PL GRIM in Algorithm 3. In each step of \mathcal{M}_{PL} , we first use Algorithm 2 to generate a full ranking γ , which is a linear extension of preference V . Then, with probability $\min \left\{ \frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})}, 1 \right\}$, let $\beta = \gamma$, otherwise the next ranking remains current ranking β .

Before analyzing the mixing time of \mathcal{M}_{PL} , we first show the probability $\frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})}$ used in step 5 of Algorithm 2 can be easily computed after GRIM sampling in the following Proposition.

Proposition 3 $\frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})} = \prod_{t=1}^{m-1} \frac{Z_{V, \gamma^{(t)}, \vec{\theta}}}{Z_{V, \beta^{(t)}, \vec{\theta}}}$.

Algorithm 3: Tuned Plackett-Luce GRIM by Markov Chain \mathcal{M}_{PL}

- 1 **Input:** Plackett-Luce parameters $\vec{\theta} = (\theta_1, \dots, \theta_m)$, partial order V_j and number of iteration I .
 - 2 **Initialization:** Generate an linear extension β over alternative set $\mathcal{E}\mathcal{S}_j$ by Algorithm 2.
 - 3 **for** $\ell = 1; \ell \leq I$ **do**
 - 4 Generate another linear extension γ using parameters $\vec{\theta}$ over alternative set $\mathcal{E}\mathcal{S}_j$ using Algorithm 2.
 - 5 With probability $\min \left\{ \frac{\text{Pr}_{PL}(\gamma|\vec{\theta})}{\text{Pr}_{PL}(\beta|\vec{\theta})} \cdot \frac{\text{Pr}_{GRIM}(\beta|\vec{\theta})}{\text{Pr}_{GRIM}(\gamma|\vec{\theta})}, 1 \right\}$,
 let $\beta = \gamma$.
 - 6 **end**
 - 7 Insert all remaining alternatives in $C \setminus \mathcal{E}\mathcal{S}_i$ to β using Algorithm 2.
 - 8 **return** R .
-

Proof: By Theorem 2, we have,

$$\begin{aligned} & \frac{\text{Pr}_{PL}(\gamma) \cdot \text{Pr}_{GRIM}(\beta)}{\text{Pr}_{PL}(\beta) \cdot \text{Pr}_{GRIM}(\gamma)} \\ &= \frac{\frac{1_{\gamma \in \text{Ext}(V)}}{Z_V} \cdot \text{Pr}_{PL}(\gamma|\vec{\theta}, V)}{\frac{1_{\beta \in \text{Ext}(V)}}{Z_V} \cdot \text{Pr}_{PL}(\beta|\vec{\theta}, V)} \cdot \frac{\frac{1_{\beta \in \text{Ext}(V)}}{\prod_{t=1}^{m-1} Z_{V, \beta^{(t)}, \vec{\theta}}} \cdot \text{Pr}_{PL}(\beta|\vec{\theta}, V)}{\frac{1_{\gamma \in \text{Ext}(V)}}{\prod_{t=1}^{m-1} Z_{V, \gamma^{(t)}, \vec{\theta}}} \cdot \text{Pr}_{PL}(\gamma|\vec{\theta}, V)} \\ &= \prod_{t=1}^{m-1} \frac{Z_{V, \gamma^{(t)}, \vec{\theta}}}{Z_{V, \beta^{(t)}, \vec{\theta}}}, \end{aligned}$$

where all normalization factors, $Z_{V, \beta^{(t)}, \vec{\theta}}$ and $Z_{V, \gamma^{(t)}, \vec{\theta}}$ are already calculated in GRIM process. \square

We now prove that the mixing time of \mathcal{M}_{PL} is exponentially upper bounded for a large class of profiles, called layered-structure profiles, defined as follows. Intuitively, alternatives in such a partial order can be partitioned into multiple layers, such that each layer contains alternatives among which no comparisons are available, and there is a linear order over layers such that alternatives in a high-ranked layer is always preferred to alternatives in a low-ranked layer.

Definition 3 (Layered-Structure Preferences) We call agent j 's preference is layered-structured if and only if there exist non-overlapping sets of alternatives C_1, \dots, C_l such that:

- 1° For any $l' \in [l]$ and any two alternatives $c_{i_1}, c_{i_2} \in C_{l'}$, no preference between c_{i_1}, c_{i_2} are given by agent j .
- 2° For any two alternatives $c_{i_1} \in C_{l_1}$ and $c_{i_2} \in C_{l_2}$, $c_{i_1} \succ_j c_{i_2}$ if and only if $l_1 < l_2$.
- 3° $\cup_{l' \in [l]} C_{l'} = \mathcal{E}\mathcal{S}_j$.

Theorem 4 For layered-structured preferences, the mixing time of Markov Chain \mathcal{M}_{PL} is $O(\eta^{m^*} \ln \epsilon^{-1})$, where $\eta = \frac{\max_{i \in [m]} \theta_i}{\min_{i \in [m]} \theta_i}$ and $m^* = |\mathcal{E}\mathcal{S}_j|$.

Proof: Since the proposal state γ is independent to current state β , we have the following Lemma:

Lemma 4.1 [Example 2 of (Liu 1996)] The mixing time of \mathcal{M}_{PL} is $O(s_{\max} \ln \epsilon^{-1})$, where $s_{\max} = \max_{\gamma \in \text{Ext}(V_i)} \frac{\text{Pr}_{PL}(\gamma|\vec{\theta})}{\text{Pr}_{GRIM}(\gamma|\vec{\theta})}$.

Armed with Lemma 4.1, we only need to provide an upper bound of s_{\max} (shown in Lemma 4.2).

Lemma 4.2 For any layered-structured preference V , $s_{\max} = \max_{\gamma \in \text{Ext}(V_i)} \frac{\text{Pr}_{PL}(\gamma|\vec{\theta})}{\text{Pr}_{GRIM}(\gamma|\vec{\theta})} \leq \eta^{m^*}$.

Proof: We use $m_{l'}$ to denote the number of alternatives in $C_{l'}$ and we define $s_{\min} = \min_{\gamma \in \text{Ext}(V_i)} \frac{\text{Pr}_{PL}(\gamma|\vec{\theta})}{\text{Pr}_{GRIM}(\gamma|\vec{\theta})}$ similar to s_{\max} . Since s_{\max} and s_{\min} are defined as maximum (minimum) ratio between the probabilities of GRIM and Plackett-Luce, we have $s_{\max} \geq 1 \geq s_{\min}$. According to Proposition 3, we have,

$$s_{\max} \leq \frac{s_{\max}}{s_{\min}} = \max_{\gamma, \beta \in \text{Ext}(V)} \left(\prod_{t=1}^{m-1} \frac{Z_{V, \gamma^{(t)}, \vec{\theta}}}{Z_{V, \beta^{(t)}, \vec{\theta}}} \right). \quad (6)$$

Inequality (6) reduce the problem to bounding GRIM normalization factors. Next, we will focus on layered-structure preferences and show the their normalization factors are bounded as:

$$\frac{Z_{V, \gamma^{(t-1)}, \vec{\theta}}}{Z_{V, \beta^{(t-1)}, \vec{\theta}}} \leq \begin{cases} 1 & t < m_1 \\ \eta^t & t \geq m_1 \end{cases}, \quad (7)$$

where $\gamma, \beta \in \text{Ext}(V)$ are two extensions of V .

W.l.o.g., we let the insertion order in GRIM follow the same order of $C_1 \rightarrow \dots \rightarrow C_l$. Recalling the justification of RIM algorithm, we know GRIM is a perfect sampler on C_1 since all alternatives in C_1 can be inserted to any vacancies and the case of $t < m_1$ follows.

Recalling the definition of $Z_{V, \gamma^{(t-1)}, \vec{\theta}}$, alternative $c_t \in C_{l'}$'s normalization factor $Z_{V, \gamma^{(t-1)}, \vec{\theta}}$ is $\sum_{i=1}^t \mathbf{1}_{\gamma_i^{(t)} \in \text{Ext}(\gamma^{(t-1)})}$.

$$\frac{\text{Pr}_{PL}(\gamma_i^{(t)} | (\theta_{1,1}, \dots, \theta_{1,m_1}, \dots, \theta_{l',1}, \dots, \theta_{l',t-(m_1+\dots+m_{l'-1})}))}{\text{Pr}_{PL}(\gamma_i^{(t)} | (\theta_{1,1}, \dots, \theta_{1,m_1}, \dots, \theta_{l',1}, \dots, \theta_{l',t-(m_1+\dots+m_{l'-1})-1})},$$

where $\theta_{i,j}$ represent the j -th inserted alternative in C_i . Since all alternatives in $C_{l'}$ are strictly less preferred to alternatives in $C_1, \dots, C_{l'-1}$, only the last $t - \sum_{i \in [l'-1]} m_i$ vacancies are allowed by linear extension condition, and we have,

$$\begin{aligned} & \frac{Z_{V, \gamma^{(t-1)}, \vec{\theta}}}{Z_{V, \beta^{(t-1)}, \vec{\theta}}} \\ &= \sum_{i=m_1}^t \frac{\text{Pr}_{PL}(\gamma_i^{(t)} | (\theta_{1,1}, \dots, \theta_{1,m_1}, \dots, \theta_{l',1}, \dots, \theta_{l',t-(m_1+\dots+m_{l'-1})}))}{\text{Pr}_{PL}(\gamma_i^{(t)} | (\theta_{1,1}, \dots, \theta_{1,m_1}, \dots, \theta_{l',1}, \dots, \theta_{l',t-(m_1+\dots+m_{l'-1})-1})} \\ &= \left(\prod_{j=1}^{\sum_{i \in [l'-1]} m_i} \frac{\theta_{\gamma_j}}{\theta_t + \sum_{i'=j}^{t-1} \theta_{\gamma_{i'}}} \right) \cdot \left(\prod_{j=1}^{t-1} \frac{\theta_{\gamma_j}}{\sum_{i'=j}^{t-1} \theta_{\gamma_{i'}}} \right), \end{aligned}$$

where γ_j denote the j -th ranked alternative in $\gamma^{(t-1)}$. Doing the same deviation to ranking $\beta^{(t-1)}$ and combine it with the result of γ , for any alternative $c_t \in C_{l'}$ we have,

$$\begin{aligned} & \frac{Z_{V, \gamma^{(t-1)}, \vec{\theta}}}{Z_{V, \beta^{(t-1)}, \vec{\theta}}} \\ &= \left(\prod_{j=1}^{\sum_{i \in [l'-1]} m_i} \frac{\theta_t + \sum_{i'=j}^{t-1} \theta_{\beta_{i'}}}{\theta_t + \sum_{i'=j}^{t-1} \theta_{\gamma_{i'}}} \right) \cdot \left(\prod_{j=1}^{t-1} \frac{\sum_{i'=j}^{t-1} \theta_{\beta_{i'}}}{\sum_{i'=j}^{t-1} \theta_{\gamma_{i'}}} \right) \\ &\leq \left(\prod_{j=1}^{\sum_{i \in [l'-1]} m_i} \frac{\theta_{\max}}{\theta_{\min}} \right) \cdot \left(\prod_{j=1}^{t-1} \frac{\theta_{\max}}{\theta_{\min}} \right) \leq \eta^t. \end{aligned}$$

Lemma 4.2 follows by combining (7) and (6). \square

Theorem 4 follows by applying Lemma 4.2 to Lemma 4.1. \square

The Gibbs Sampler for Plackett-Luce Model

We first introduce one exact Plackett-Luce sampler for full rankings using random utility interpretation (Theorem 5). It is noteworthy that the complexity of this sampler is $O(m)$ per iteration (or equivalently, the complexity of one Gibbs iteration in Algorithm 4 is $O(m^*)$).

Algorithm 4: Gibbs Plackett-Luce Sampler

- 1 **Input:** Plackett-Luce parameter $\vec{\theta} = (\theta_1, \dots, \theta_m)$, partial order V_j and number of iterations I .
 - 2 **Initialization:** Arbitrarily set an $1 \times m$ array of utilities $u = (u_1, \dots, u_m)$ and set of inserted alternatives $S = \emptyset$.
 - 3 **for** $\ell = 1; \ell \leq I$ **do**
 - 4 **for** $c_t \in \mathcal{ES}_j$ **do**
 - 5 Calculate $l_U = \min_{i \in U_i} e^{-\theta_i \cdot e^{-u_i}}$, where $U_i = \{i \mid (c_i \in S) \wedge ([c_i \succ_j c_t] \in V_j)\}$.
 - 6 Calculate $l_L = \max_{i \in L_i} e^{-\theta_i \cdot e^{-u_i}}$, where $L_i = \{i \mid (c_i \in S) \wedge ([c_t \succ_j c_i] \in V_j)\}$.
 - 7 Uniformly generate a random number r from interval $(\max\{0, l_L\}, \min\{1, l_U\})$.
 - 8 Let $u_t = -\ln(-\ln r + \ln \theta_t)$ and add c_t to S .
 - 9 **end**
 - 10 **end**
 - 11 Sample the remaining alternatives' utilities according to Theorem 5.
 - 12 Sort the utilities (u_1, \dots, u_m) to get the linear order R .
 - 13 **return** R .
-

Theorem 5 (Yellott 1977, Theorem 5) For sample space $S = \mathcal{L}(C)$ and parameters $\vec{\theta} = (\theta_1, \dots, \theta_m)$, Plackett-Luce sampling is equivalent to sample from Gumbel distributions whose cumulative distribution functions (CDFs) are $e^{-\theta_i \cdot e^{-x}}$.

Our Gibbs Plackett-Luce sampler given arbitrary partial orders (Algorithm 4) is based on the PL sampler introduced above and the Gibbs sampler by Azari Soufiani, Parkes, and Xia (2012). We let $l_U = 1$ (or $l_L = 0$) when $U_i = \emptyset$ (or $L_i = \emptyset$). We note that the sampler shown in Algorithm 4 is applicable to any Random Utility Models with given CDF on utility distributions.

Remark for Gibbs Sampler Bounding the mixing time of Gibbs sampler for arbitrary (truncated) distribution is still an open question. Moreover, the mixing time of Algorithm 4 is even harder to bound because it focuses on the distribution over rankings instead of utilities. Nevertheless, Gibbs sampler might be a fast algorithm in experiments (or, averaged case) because it takes the advantage of lower complexity per iteration.

Experiments

We illustrate the performances of our algorithms on both synthetic data and real-world data. We note that there is no existing algorithm for learning Plackett-Luce model and its mixtures from general partial orders without discarding any

information. All experiments with recorded runtime were run on an Ubuntu Linux server with Intel Xeon E5 v3 CPUs each clocked at 3.50 GHz.

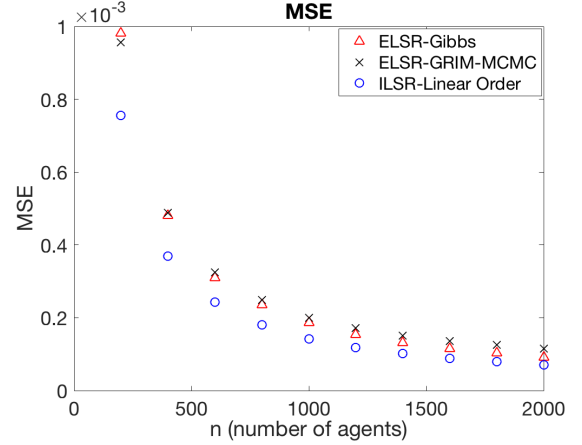


Figure 3: MSE of the proposed EM algorithms for a single Plackett-Luce model.

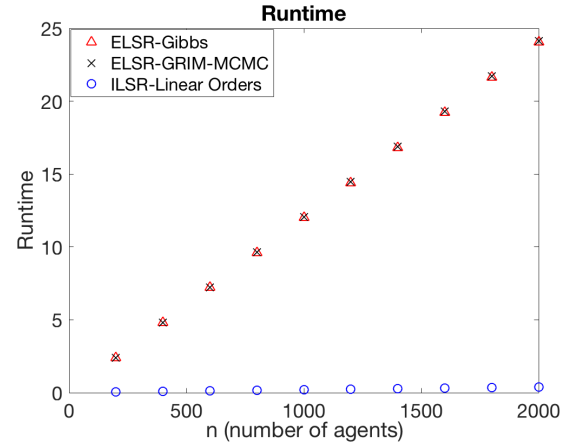


Figure 4: Runtime (in seconds) of the proposed EM algorithms for a single Plackett-Luce model. Note ILSR-Linear Orders is **not** applicable to arbitrary partial preferences.

Experiments on Synthetic Data

Learning the Plackett-Luce Model from Partial Orders.

We first generated linear orders from the Plackett-Luce model over 10 alternatives. The ground truth parameters $\vec{\theta} = (\theta_1, \dots, \theta_m)$ were generated uniformly at random and normalized s.t. $\sum_{i=1}^m \theta_i = 1$. The linear orders were sampled by Definition 1. Then we sampled partial orders in the following way. We fixed $p = 0.5$. Given any linear order, we sampled from all $\frac{m(m-1)}{2}$ pairwise comparisons by keeping each pairwise comparison with probability p independently. The sampled partial order is then converted to its transitive closure. In average, below 75% of all pairwise comparisons were sampled. We ran our proposed EM algorithms with

GRIM-MCMC and Gibbs samplers respectively, denoted by “ELSR-GRIM-MCMC” and “ELSR-Gibbs”. Two linear extensions were sampled for each partial order. We use five EM iterations for each algorithm. Values were averaged over 2000 trials. To show the statistical efficiency of our algorithms, we also learned from linear orders (which are assumed to be unobservable) using the ILSR algorithm (denoted by “ILSR-Linear Order”) by (Maystre and Grossglauser 2015), which is the iterative LSR. Results are shown in Figures 3 and 4. **Observations.** Both statistical efficiency and computational efficiency of the algorithm with Gibbs sampler are slightly better than that with GRIM sampler. The statistical efficiency of both our algorithms are close to ILSR, which learns from supposedly unobservable linear orders.

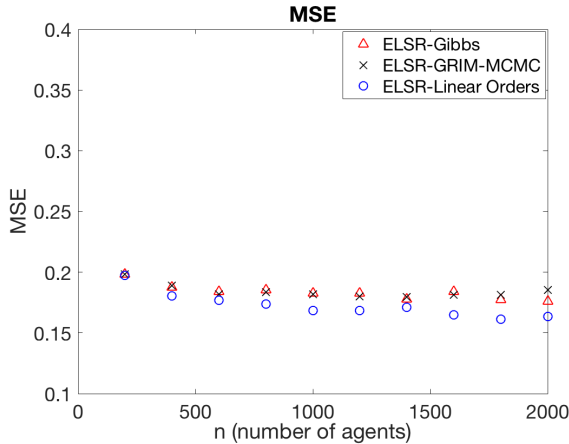


Figure 5: MSE of the proposed EM algorithms for mixtures of 3 Plackett-Luce models.

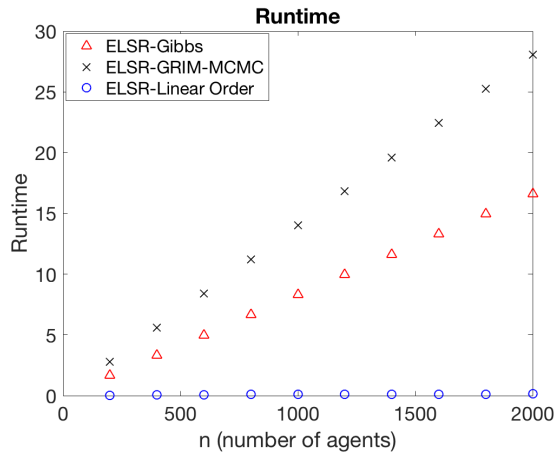


Figure 6: Runtime (in seconds) of the proposed EM algorithms for mixtures of 3 Plackett-Luce models.

Learning Mixtures of Plackett-Luce Models. The setup is similar to the previous experiments. We generated the ground truth parameters for 3-PL over 6 alternatives by generating the mixing coefficients and the parameter of each component

uniformly at random and normalizing s.t. $\sum_{r=1}^k \alpha_r = 1$ and for all r , $\sum_{i=1}^m \theta_i^{(r)} = 1$. Then we sampled partial orders using exactly the same way with a fixed $p = 0.5$. In average, below 65% of all pairwise comparisons were sampled due to fewer alternatives. 6 linear extensions were generated from each partial order. We use five EM iterations for each algorithm. Values were averaged over 2000 trials. **Observations.** From Figures 5 and 6 we observe that our algorithm with Gibbs sampler is more computationally efficient than that with GRIM-MCMC sampler. Again, both algorithms have similar statistical efficiency “ELSR-Linear Order”, which learns from supposedly unobservable linear orders.

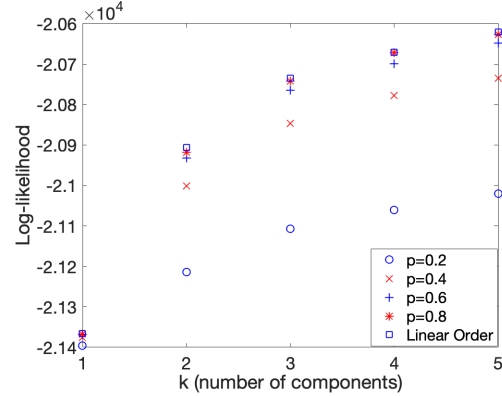


Figure 7: Log-likelihood of sushi test data as k increases when a k -PL is learned from the training data. Values were averaged over 100 trials.

Experiments on Real-World Data

The sushi data from Preflib (Mattei and Walsh 2013) consists of 5000 linear orders of 10 types of sushi. We randomly split this dataset into training data (3500 linear orders) and test data (1500 partial orders). To generate partial orders, we sample pairwise comparisons with $p = 0.2, 0.4, 0.6, 0.8, 1$ and learn a k -PL from the partial orders using the proposed E-LSR algorithm with Gibbs sampler, where k ranges from 1 to 5. We measure the fitness of a k -PL using the likelihood of the test data given the learned parameter. Results are shown in Figure 7. **Observations.** We observe that when p increases, the learning outcome fits the test data better. For $k \leq 5$, k -PL fits the test data better as k increases.

Conclusions and Future Work

We propose an EM-based framework with two samplers for learning Plackett-Luce model and its mixtures from partial orders. We demonstrate the efficiency of our algorithms with experiments on both synthetic datasets and real-world data and conclude that Gibbs sampler outperforms GRIM-MCMC. For future work we plan to explore faster algorithms to sample linear extensions from partial orders or other approaches to learn mixtures of Plackett-Luce models, as well as other mixture models.

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

- 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.
- Baltrunas, L.; Makcinskis, T.; and Ricci, F. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, 119–126. ACM.
- Doignon, J.-P.; Pekeč, A.; and Regenwetter, M. 2004. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika* 69(1):33–54.
- Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. 2001. Rank aggregation methods for the web. In *Proceedings of the 10th World Wide Web Conference*, 613–622.
- Ford Jr, L. R. 1957. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly* 64(8P2):28–33.
- 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.
- Hastings, W. K. 1970. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika* 57(1):97–109.
- Hüllermeier, E.; Fürnkranz, J.; Cheng, W.; and Brinker, K. 2008. Label ranking by learning pairwise preferences. *Artificial Intelligence* 172(16-17):1897–1916.
- Hunter, D. R. 2004. MM algorithms for generalized Bradley-Terry models. In *The Annals of Statistics*, volume 32, 384–406.
- Khetan, A., and Oh, S. 2016a. Computational and statistical tradeoffs in learning to rank. In *Advances in Neural Information Processing Systems*, 739–747.
- Khetan, A., and Oh, S. 2016b. Data-driven rank breaking for efficient rank aggregation. *Journal of Machine Learning Research* 17(193):1–54.
- Liu, A.; Wu, Q.; Zhenming, L.; and Xia, L. 2019. Near-neighbor methods in random preference completion. In *Proceedings of 33rd AAAI Conference on Artificial Intelligence (AAAI-19)*.
- Liu, J. S. 1996. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* 6(2):113–119.
- Liu, T.-Y. 2011. *Learning to Rank for Information Retrieval*. Springer.
- Lu, T., and Boutilier, C. 2014. Effective sampling and learning for mallows models with pairwise-preference data. *The Journal of Machine Learning Research* 15(1):3783–3829.
- 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.
- Maystre, L., and Grossglauer, M. 2015. Fast and accurate inference of Plackett–Luce models. In *Advances in Neural Information Processing Systems*, 172–180.
- Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* 21(6):1087–1092.
- 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. 2017. Rank centrality: Ranking from pairwise comparisons. *Operations Research* 65(1):266–287.
- Oh, S., and Shah, D. 2014. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, 595–603.
- Plackett, R. L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 24(2):193–202.
- 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.
- Yellott, J. I. J. 1977. The relationship between Luce’s Choice Axiom, Thurstone’s Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology* 15(2):109–144.
- Zhao, Z., and Xia, L. 2018. Composite marginal likelihood methods for random utility models. In *Proceedings of The 35th International Conference on Machine Learning*.
- Zhao, Z.; Li, H.; Wang, J.; Kephart, J.; Mattei, N.; Su, H.; and Xia, L. 2018. A cost-effective framework for preference elicitation and aggregation. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*.
- Zhao, Z.; Piech, P.; and Xia, L. 2016. Learning mixtures of Plackett-Luce models. In *Proceedings of The 33rd International Conference on Machine Learning*.
- Zhao, Z.; Villamil, T.; and Xia, L. 2018. Learning mixtures of random utility models. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence (AAAI-18)*.