

Effective Sampling and Learning for Mallows Models with Pairwise-Preference Data

Tyler Lu

Craig Boutilier

Department of Computer Science

University of Toronto

6 King's College Rd.

Toronto, ON, Canada M5S 3G4

TL@CS.TORONTO.EDU

CEBLY@CS.TORONTO.EDU

Editor: Guy Lebanon

Abstract

Learning preference distributions is a critical problem in many areas (e.g., recommender systems, IR, social choice). However, many existing learning and inference methods impose restrictive assumptions on the form of user preferences that can be admitted as evidence. We relax these restrictions by considering as data arbitrary *pairwise comparisons* of alternatives, which represent the fundamental building blocks of ordinal rankings. We develop the first algorithms for learning Mallows models (and mixtures thereof) from pairwise comparison data. At the heart of our technique is a new algorithm, the *generalized repeated insertion model (GRIM)*, which allows sampling from arbitrary ranking distributions, and conditional Mallows models in particular. While we show that sampling from a Mallows model with pairwise evidence is computationally difficult in general, we develop approximate samplers that are exact for many important special cases—and have provable bounds with pairwise evidence—and derive algorithms for evaluating log-likelihood, learning Mallows mixtures, and non-parametric estimation. Experiments on real-world data sets demonstrate the effectiveness of our approach.¹

Keywords: preference learning, ranking, incomplete data, Mallows models, mixture models

1. Introduction

With the abundance of preference data from search engines, review sites, etc., there is tremendous demand for learning detailed models of user preferences to support personalized recommendation, information retrieval, social choice, and other applications. Much work has focused on ordinal preference models and learning user or group *rankings* of alternatives or items. Within this setting, we can distinguish two classes of models. First, we may wish to learn an underlying *objective* (or “correct”) ranking from noisy data or noisy expressions of user preferences (e.g., as in web search, where user selection suggests relevance), a view adopted frequently in IR and “learning to rank” (Burges, 2010) and occasionally in social choice (Young, 1995). Second, we might assume that users have different

1. Some parts of this paper appeared in: T. Lu and C. Boutilier, Learning Mallows Models with Pairwise Preferences, *Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML 2011)*, pp.145-152, Bellevue, WA (2011).

types with inherently distinct preferences, and learn a population model that explains this diversity. Learning preference types (e.g., by segmenting or clustering the population) is key to effective personalization and preference elicitation in recommender systems, social choice, and numerous other domains. For example, with a learned population preference distribution, choice data obtained from a specific user allows inferences to be drawn about her preferences. In this work, we focus on the latter setting, learning preference distributions when users have genuinely distinct preferences.

Considerable work in machine learning has exploited ranking models developed in the statistics and psychometrics literature, such as the Mallows model (Mallows, 1957), the Plackett-Luce model (Plackett, 1975; Luce, 1959), and others (Marden, 1995), as well as their non-parametric representations (Lebanon and Mao, 2008). However, most research to date provides methods for learning preference distributions using very restricted forms of evidence about individual user preferences, whether passively observed or actively elicited, ranging from complete rankings, to top- t /bottom- t alternatives, to *partitioned preferences* (Lebanon and Mao, 2008). Missing from this list are arbitrary *pairwise comparisons* of the form “alternative a is preferred to alternative b .” Such pairwise preferences form the building blocks of almost all reasonable evidence about preferences, and subsumes the most general evidential models proposed in the literature. Furthermore, preferences in this form naturally arise in active elicitation of user preferences and choice contexts (e.g., web search, product comparison, advertisement clicks), where a user selects one alternative over others in some set (Louviere et al., 2000). In general, data about a user’s preferences will often take the form of arbitrary choice sets as is common in web search, online advertising, product comparison, etc. But none of the techniques and algorithms developed to date can learn from such choice sets. These preferences can be as simple as a single paired comparison: “I like alternative a better than b ,” or as complex as a set of comparisons: “I like a better than b, c, \dots , and I like z better than y, x, \dots ” In this sense, pairwise comparisons should be viewed as the fundamental building block and universal language of ordinal preference ranking.²

While learning with pairwise preferences is clearly of great importance, it is widely believed that learning probabilistic models of ordinal preference using paired comparison data is impractically difficult (indeed, we show this formally below). As a consequence, the Mallows model is often shunned in favor of more inference-friendly models (e.g., Plackett-Luce, which accommodates more general, but still restrictive, preferences; see Cheng et al., 2010; Guiver and Snelson, 2009). To date, no methods have been proposed for learning from arbitrary pairwise preferences in any of the commonly used ranking models in machine learning. We tackle this problem directly by developing techniques for learning Mallows models, and mixtures thereof, from pairwise preference data.

Our core contribution is the *generalized repeated insertion model (GRIM)*, a new method for sampling from *arbitrary ranking distributions*—including conditional Mallows—that generalizes the repeated insertion method for unconditional sampling of Mallows models (Doignon et al., 2004). We show that even evaluating the log-likelihood under a Mallows model with respect to arbitrary ordinal data is #P-hard, implying that learning will be at

2. Of course, ordinal preferences do not capture strength of preference; but real-valued or scaled preferences (e.g., movie or book ratings) can be converted to pairwise preferences readily, albeit with some loss of information.

least as difficult. However, we derive another method, which we call AMP, which efficiently, though approximately, samples from any conditional Mallows distribution given arbitrary pairwise evidence. Moreover, we show that AMP is *exact* for important classes of evidence (including partitioned preferences), and that empirically it provides very close approximations given general pairwise evidence. We use this sampler as the core of a Monte Carlo EM algorithm to learn Mallows mixtures, evaluate log-likelihood, and make predictions about missing preferences. We also extend the non-parametric framework of Lebanon and Mao (2008) to handle unrestricted ordinal preference data. Experiments show our algorithms can effectively learn Mallows mixtures, with reasonable running time, on data sets with hundreds of alternatives and thousands of users. Our sampling algorithm can be adapted rather easily to other models as well (e.g., we show how a simple modification allows sampling from Mallows models with a weighted Kendall-tau metric).

The remainder of the paper is organized as follows. In Section 2 we describe the necessary background on ordinal preferences, Mallows models, and the *repeated insertion method* (Doignon et al., 2004) for Mallows distributions, which we extend later in the paper. We also discuss related work on learning probabilistic preference models. We introduce our main technical tool, the *generalized repeated insertion method (GRIM)*, in Section 3. We show how it can be used to sample from Mallows mixtures conditioned on incomplete preferences by first defining an approximate, but *direct* sampler AMP that is exact for important special cases, and analyzing its computational and statistical properties. We then develop Metropolis and Gibbs sampling methods that exploit AMP to soundly sample any Mallows or Mallows mixture posterior. In Section 4 we develop an EM algorithm for learning a Mallows mixture from arbitrary pairwise comparison data that leverages our sampling algorithms, and provide experimental results of this procedure on several real-world data sets in Section 5. Section 6 extends the framework of Lebanon and Mao (2008) for non-parametric estimation to handle evidence in the form of arbitrary ordinal preferences. We conclude in Section 7 with a discussion of future directions.

2. Preliminaries

We begin by describing the ordinal preferences (rankings) used in the work, providing a brief overview of several common probabilistic preference models, with an emphasis on the Mallows ϕ -model (and mixtures) and models of partial preference data. We then outline Doignon et al. (2004) repeated insertion model for sampling preferences from a Mallows distribution (and draw connections to older models for sampling rankings proposed by Condorcet, Kemeny and Young). We also briefly discuss related work on learning probabilistic preference models.

2.1 Ordinal Preferences

We assume a set of m alternatives $A = \{a_1, \dots, a_m\}$ and n agents $N = \{1, \dots, n\}$. Each agent ℓ has preferences over the set of alternatives represented by a total ordering or *ranking* \succ_ℓ over A . We write $x \succ_\ell y$ to mean ℓ prefers x to y . Rankings can be represented as permutations of A . For any positive integer b , let $[b] = \{1, \dots, b\}$. We often represent a ranking as a permutation or bijection $\sigma : A \rightarrow [m]$, where $\sigma(a)$ is the *rank* or *position* of a in the ranking. Thus, for $i \in [m]$, $\sigma^{-1}(i)$ is the alternative with rank i . We write

$\sigma = \sigma_1 \sigma_2 \cdots \sigma_m$ for a ranking with i -th ranked alternative $\sigma_i \in A$, and \succ_σ for the induced preference relation. For any $X \subseteq A$, let $\sigma|_X$ denote the ranking obtained by restricting σ to alternatives in X . Let $\mathbf{1}[\cdot]$ be the indicator function.

Generally, we do not have access to the complete preferences of agents, but only partial information about their rankings (e.g., based on choice behavior, query responses, etc.). We assume this data has a very general form: for each agent ℓ we have a set of *revealed pairwise preference comparisons* over A , or simply *preferences*:

$$v_\ell = \{x_1^\ell \succ_\ell y_1^\ell, \dots, x_{k_\ell}^\ell \succ_\ell y_{k_\ell}^\ell\}.$$

Intuitively, these reflect information about ℓ 's preferences revealed by some process. For example, this could represent product-ratings data; preference revealed by selection or purchase of certain items (e.g., web links, products) over others, or responses to survey data.

Let $\text{tc}(v_\ell)$ denote the transitive closure of v_ℓ , i.e., the smallest transitive relation containing v_ℓ . We write $\{x, y\} \in v_\ell$ if there is a comparison between x and y in v_ℓ and, similarly, $\{x, y\} \in \text{tc}(v_\ell)$ if x, y are comparable in its transitive closure. Since preferences are strict, $\text{tc}(v_\ell)$ is a strict partial order on A . We assume each v_ℓ is *consistent*, i.e., $\text{tc}(v_\ell)$ contains no cycles.³ Preferences v_ℓ are *complete* if and only if $\text{tc}(v)$ is a total order on A . Let $\Omega(v)$ denote the set of *linear extensions* of v , i.e., those rankings consistent with v . Let $\Omega = \Omega(\emptyset)$ be the set of all $m!$ complete preferences. A collection $V = (v_1, \dots, v_n)$ is a (*partial*) *preference profile*—we assume that the observed data used for inference and learning purposes in our work takes this form. Given ranking $\sigma = \sigma_1 \sigma_2 \cdots \sigma_m$ and preference v , we define the *dissimilarity* or *disagreement* $d(v, \sigma)$ between the two to be:

$$d(v, \sigma) = \sum_{i < j \leq m} \mathbf{1}[\sigma_j \succ \sigma_i \in \text{tc}(v)]. \tag{1}$$

Dissimilarity between a partial preference and a ranking is the number of pairwise disagreements among the relative ranking of alternatives, i.e., those pairs in v that are misordered relative to σ . If v is a complete ranking, $d(v, \sigma)$ is the classic *Kendall-tau* metric on rankings. Likewise, define $s(v, \sigma)$ to be the number of pairwise comparisons in $\text{tc}(v)$ that are consistent with σ . We have that $d(v, \sigma) + s(v, \sigma)$ is the number of comparisons in $\text{tc}(v)$. If v is complete, then $d(v, \sigma) + s(v, \sigma) = \binom{m}{2}$.

Arbitrary sets v of pairwise comparisons can be used to model a wide range of realistic revealed preferences:⁴

- Complete rankings require $m - 1$ paired comparisons (e.g, $a \succ b \succ c \dots$), and can be elicited with at most $m(m - 1)/2$ paired comparison queries.
- Top- t preferences (Busse et al., 2007) require that users provide a complete ranking of their top t most preferred alternatives. These can be represented using $m - 1$ pairs:

3. Many of the concepts for probabilistic modeling, inference and learning developed in this paper can be applied *mutatis mutandis* to models where revealed preferences are noisy; however, we leave this topic to future research.

4. One exception to this is information about preferences that involve “disjunctive” constraints. For instance, a response to the question “What alternative is ranked t^{th} ?” cannot be mapped to a set of pairwise preferences unless the positions t are queried in ascending or descending order (hence inducing top- t or bottom- t preferences).

$t - 1$ comparisons to order the top t alternatives, and $m - t$ pairs to ensure the t -th alternative is ranked above the remaining $m - t$. Bottom- t preferences are similar.

- Complete rankings of *subsets* $X \subseteq A$ (Guiver and Snelson, 2009; Cheng et al., 2010) are also representable in the obvious fashion (requiring $k - 1$ comparisons if $|X| = k$).
- Preferences revealed by the choice of an alternative a from $X \subseteq A$ (Louviere et al., 2000) can also be represented using $k - 1$ pairs of the form $a \succ b$ for each $b \in X \setminus \{a\}$ (where $|X| = k$). Sets of such choices are captured in the obvious way.
- Ordinal ratings data: if alternatives are scored on an ordinal scale s (e.g., a scale of 1–5 where 1 is most preferred), we simply include $a \succ b$ whenever $s(a) < s(b)$, assuming that alternatives with the same rating cannot be compared using the level of granularity provided.

Much of the existing work in learning or modelling distributions over ordinal preferences restricts the class of representable preferences. Much work has focused on top- t preferences (Busse et al., 2007; Meila and Chen, 2010; Gormley and Murphy, 2007; Fligner and Verducci, 1986, 1993), and its generalizations (Lebanon and Mao, 2008); other papers have worked with rankings of a subset of alternatives (Guiver and Snelson, 2009; Cheng et al., 2010). The main issue in allowing arbitrary consistent collections of paired preferences, which can represent all of the above special cases, is the difficult inference problem that results. The primary aim of this work is to develop tractable inference algorithms for a much broader and realistic class of preferences. Before closing our discussion of ordinal preferences, we define a recently studied and relatively expressive class of preferences

Definition 1 (Lebanon and Mao 2008) *A partial preference v is a partitioned preference if A can be partitioned into subsets A_1, \dots, A_q s.t.: (a) for all $i < j \leq q$, if $x \in A_i$ and $y \in A_j$ then $x \succ_{\text{tc}(v)} y$; and (b) for each $i \leq q$, alternatives in A_i are incomparable under $\text{tc}(v)$.*

Partitioned preferences are quite general, subsuming some of the special cases above, including top- t or bottom- t preferences, or ratings data. However, they cannot represent many naturally occurring preferences, including those as simple as a single pairwise comparison $a \succ b$. We demonstrate below that our techniques can be applied effectively to such preferences.

2.2 Mallows Models and Sampling Procedures

There are many distributional models of rankings that have been developed in psychometrics, statistics and econometrics to explain choice behavior (Marden, 1995 provides a good overview). Two of the more popular in the machine learning community are the *Mallows model* (Mallows, 1957) and the *Plackett-Luce model* (Plackett, 1975; Luce, 1959). We focus on Mallows in this work, though we believe our methods can be extended to other models.

2.2.1 THE MALLOWS MODEL

The *Mallows ϕ -model* (which we simply call the Mallows model hereafter) is typical of a wide-range of distance-based ranking models (Mallows, 1957; Marden, 1995). As above, let

d be the Kendall-tau distance. The Mallows model is parameterized by a modal or *reference ranking* σ and a *dispersion parameter* $\phi \in (0, 1]$. For any ranking r , the Mallows model specifies:

$$P(r) = P(r | \sigma, \phi) = \frac{1}{Z} \phi^{d(r, \sigma)}, \quad (2)$$

where $Z = \sum_{r' \in \Omega} \phi^{d(r', \sigma)}$ is the normalization constant. It can be shown that

$$Z = 1 \cdot (1 + \phi) \cdot (1 + \phi + \phi^2) \cdots (1 + \cdots + \phi^{m-1}). \quad (3)$$

When $\phi = 1$ we obtain the uniform distribution over Ω (in the social choice literature, this model is known as *impartial culture*). As $\phi \rightarrow 0$, the distribution concentrates all mass on σ . The model can also be expressed as $P(r | \sigma, \lambda) = \frac{1}{Z} e^{-\lambda d(r, \sigma)}$, where $\lambda = -\ln \phi \geq 0$. Various extensions and generalizations of this model have been developed (e.g., using other distance measures) (Marden, 1995).

2.2.2 CONDORCET'S DECISION PROBLEM

We describe a simple sampling procedure proposed by Mallows, Condorcet and further analyzed by Young, since this will motivate the RIM sampler discussed in Section 2.2.3. Mallows (1957) explained his model using process in which a judge assesses alternatives by repeatedly making pairwise comparisons. The outcome of such a comparison is stochastic and depends on the reference ranking σ . If x and y are compared and x is preferred to y in σ , then the judge “correctly” assesses $x \succ y$ with probability $1 - p_{xy}$, and erroneously assesses $y \succ x$ with probability $p_{xy} < 1/2$. Each assessment is independent of other comparisons. Mallows’ process generated a pairwise comparison for each pair of alternatives as described: after all paired comparisons are made, if the result is consistent (i.e., corresponds to a ranking), it is accepted; otherwise the process is repeated. While the error probability p_{xy} can depend in a fairly general way on their positions in σ , if $p_{xy} = p$ for all x, y then we obtain the Mallows model.

Such a probabilistic view of rankings was studied two centuries earlier by Nicolas de Condorcet in the context of collective political decision making (Condorcet, 1785). He modeled his view of the role of government, that of making the “right decisions,” by considering the selection from a set of choices (e.g., policies), one that maximizes benefit to society. Members of society, or *voters*, express their opinion in the form of a ranking over choices. He assumed that some (latent) *objective ranking* orders choices from most to least beneficial to society and that each voter is able to provide an independent, random assessment of relative rank of any pair of choices: if $a \succ b$, in the objective ranking a voter will assess that to be the case with probability $1 - p$, with a error probability less than $1/2$. Instead of studying the probabilistic model *per se*, Condorcet addressed the *decision problem*: how to find the ranking most likely to be correct. For the case of three alternatives, he proved that the ranking which minimized the total number pairwise preference disagreements (i.e., Kendall-tau distance) with respect to the stated voter rankings was the most likely to be correct.

In modern parlance, Condorcet showed how to compute the maximum likelihood estimator (MLE) of the objective or reference ranking. Kemeny (1959) proposed the *Kemeny ranking* as a general method for aggregating noisy voter rankings, extending Condorcet’s

approach to accommodate any number of alternatives. The Kemeny ranking is that which minimizes total number of pairwise preference disagreements with the set of voter rankings, which Kemeny justified axiomatically (showing it to be the only aggregate ranking that satisfies certain intuitive axioms). A statistical rationale for Kemeny’s approach was provided by Young (1995), who extended Condorcet’s analysis, showing that, for *any* number of alternatives, under Condorcet’s noise model, the MLE of the reference ranking is in fact the Kemeny ranking. These two independent threads (Condorcet-Kemeny-Young and Mallows) can both be viewed as statistical estimation of a noisy ranking model. We tie these threads together, showing that Condorcet’s noise model for any number of alternatives corresponds to the Mallows models (which implies, by Young’s result, that the Kemeny ranking is the MLE for the Mallows model). The Condorcet-Mallows noisy ranking process can be formalized as follows:

Pairwise Comparison Sampling of Mallows

1. Let σ be the reference ranking and $0 \leq p \leq 1/2$.
2. Initialize $v \leftarrow \emptyset$.
3. For each pair of items x, y in A , such that $x \succ_{\sigma} y$,
 - (a) with probability $1 - p$ add $x \succ y$ to v ,
 - (b) otherwise add $y \succ x$ to v .
4. If v is intransitive, go back to step 1 and start over.
5. v is transitive and corresponds to a ranking.

This pairwise comparison process generates rankings in accordance with the Mallows model (Equation 2), a fact shown by Mallows (1957), but which we derive here (since it will be instructive below). Consider the following distribution over rankings v :

$$P'(v \mid \sigma, p) = \frac{1}{Z'} \prod_{\{x,y\} \subseteq A} \begin{cases} p & \text{if } v \text{ and } \sigma \text{ disagree on } x, y \\ 1 - p & \text{otherwise,} \end{cases} \tag{4}$$

where Z' is the normalization constant (i.e., the sum of the probabilities generated by the above procedure, over all transitive, complete preferences). The form of this distribution corresponds exactly to the rankings generated. This can be seen by noticing that the generating procedure independently decides for each pair of alternatives x, y , with a flip of p -biased coin, whether to order them according to σ . Since intransitive preferences v are discarded by the procedure, the generating procedure corresponds to P' . We can simplify the expression for P' to:

$$\begin{aligned} P'(v \mid \sigma, p) &= \frac{1}{Z'} p^{d(v,\sigma)} (1 - p)^{s(v,\sigma)} \\ &= \frac{1}{Z'} p^{d(v,\sigma)} (1 - p)^{\binom{m}{2} - d(v,\sigma)} \\ &= \frac{1}{Z'} (1 - p)^{\binom{m}{2}} \left(\frac{p}{1 - p} \right)^{d(v,\sigma)}. \end{aligned} \tag{5}$$

By setting $\phi = \frac{p}{1-p}$, recalling the definition of Z (Equation 3), and noticing that

$$Z' = (1-p)^{\binom{m}{2}} Z \tag{6}$$

$$= (1-p)^{\binom{m}{2}} \left(1 + \frac{p}{1-p}\right) \left(1 + \frac{p}{1-p} + \left(\frac{p}{1-p}\right)^2\right) \cdots \left(1 + \cdots + \left(\frac{p}{1-p}\right)^{m-1}\right), \tag{7}$$

we obtain Equation 2. The log-likelihood, given observed complete rankings r_1, \dots, r_n , is

$$\sum_{\ell=1}^n [d(r_\ell, \sigma) \ln \phi - \ln Z].$$

Hence, the MLE ranking is the minimizer of $\sum_{\ell=1}^n d(r_\ell, \sigma)$, namely, the Kemeny ranking.

2.2.3 THE REPEATED INSERTION MODEL

The Condorcet/Mallows sampling procedure for drawing rankings from the Mallows distribution can be very inefficient, since it relies on rejection of partially constructed rankings as soon as a single circular, or non-transitive, triad ($a \succ b \succ c \succ a$) is drawn. While the original motivation for these models was not computational, efficient sampling is important for a variety of inference and learning tasks. Doignon et al. (2004) introduce the *repeated insertion model (RIM)* for the analysis of probabilistic models of approval voting, but which also provides a much more effective means of sampling from a Mallows distribution.

RIM is a generative process that gives rise to a family of distributions over rankings and provides a practical way to sample rankings from a Mallows model. The model assumes some reference ranking $\sigma = \sigma_1 \sigma_2 \cdots \sigma_m$, and *insertion probabilities* p_{ij} for each $i \leq m, j \leq i$. RIM generates a new *output ranking* using the following process, proceeding in m steps. At step 1, σ_1 is added to the output ranking. At step 2, σ_2 is inserted above σ_1 with probability $p_{2,1}$ and inserted below with probability $p_{2,2} = 1 - p_{2,1}$. More generally, at the i -th step, the output ranking will be an ordering of $\sigma_1, \dots, \sigma_{i-1}$ and σ_i will be inserted at rank $j \leq i$ with probability p_{ij} . Critically, the insertion probabilities are *independent of the ordering of the previously inserted alternatives*.

It is easy to see that one can generate any ranking with the appropriate insertion positions. As we describe below, Doignon et al. (2004) show that one can sample from a Mallows distribution using RIM with appropriate insertion probabilities. We now introduce several concepts that can be used to more easily formalize and analyze RIM, and our subsequent extensions of it.

Definition 2 *Let $\sigma = \sigma_1 \cdots \sigma_m$ be a reference ranking. Let an insertion vector be any positive integer vector $\mathbf{j} = (j_1, \dots, j_m)$ satisfying $j_i \leq i, \forall i \leq m$; and let I be the set of such insertion vectors. A repeated insertion function $\Phi_\sigma : I \rightarrow \Omega$ maps an insertion vector \mathbf{j} into a ranking $\Phi_\sigma(\mathbf{j})$ by placing each σ_i , in turn, into rank j_i , for all $i \leq m$.*

This definition is best illustrated with an example. Consider the insertion vector $(1, 1, 2, 3)$ and reference ranking $\sigma = abcd$. In this case, $\Phi_\sigma(1, 1, 2, 3) = bcda$ because: we first insert a into rank 1; we then insert b into rank 1, shifting a down to obtain partial ranking ba ; we then insert c into rank 2, leaving b in place, but moving a down, obtaining bca ; finally,

we insert d at rank 3, giving $bcda$. By the same process we obtain $\Phi_\sigma(1, 2, 3, 4) = abcd$, and $\Phi_\sigma(1, 1, 1, 1) = dcba$. Given reference ranking σ , there is a one-to-one correspondence between rankings and insertion vectors.

Observation 3 *For any reference ranking σ , the repeated insertion function Φ_σ is a bijection between I and Ω .*

Sampling using RIM can be characterized as follows:

Definition 4 *The repeated insertion model is a probabilistic model over rankings defined by a reference ranking σ , the repeated insertion function $\Phi_\sigma(j_1, \dots, j_m)$ and a sequence of insertion probabilities p_{ij_i} for $i \leq m$, $j_i \leq i$, such that $\sum_{j=1}^i p_{ij} = 1$, $\forall i \leq m$. A ranking is generated at random by first drawing an insertion vector $\mathbf{j} = (j_1, \dots, j_m) \in I$, where each j_i is drawn independently with probability p_{ij_i} , and then applying the insertion function $\Phi_\sigma(\mathbf{j})$.*

Let $\Phi_\sigma^{-1}(r) = (j'_1, \dots, j'_m)$. Then the probability of generating a particular ranking r under RIM is $\prod_{i \leq m} p_{ij'_i}$. It is easy to see that the Kendall-tau distance between the reference ranking and the ranking induced by an insertion vector is the sum of the number “insertion misorderings” over all alternatives:

Proposition 5 *For any insertion vector $\mathbf{j} = (j_1, \dots, j_m) \in I$, we have that*

$$\sum_{i=1}^m i - j_i = d(\Phi_\sigma(\mathbf{j}), \sigma). \tag{8}$$

Proof Observe that whenever σ_i is inserted at the j_i -th position, it creates $i - j_i$ pairwise misorderings with respect to alternatives $\sigma_1, \dots, \sigma_{i-1}$. All pairwise misorderings can be accounted for this way. Summing over all $i \leq m$ gives the Kendall-tau distance. ■

Doignon et al. (2004) show that by setting the insertion probabilities p_{ij} appropriately, the resulting generative process corresponds to the Mallows model. We reprove their Theorem here, since the proof will be instructive later.

Theorem 6 (Doignon et al. 2004) *By setting insertion probabilities $p_{ij} = \phi^{i-j}/(1 + \phi + \dots + \phi^{i-1})$ for $j \leq i \leq m$, the distribution induced by RIM with insertion function Φ_σ is identical to that of the Mallows model with reference ranking σ and dispersion parameter ϕ .*

Proof We reprove the Doignon et al. (2004) theorem. Let r be any ranking and σ the reference ranking of the Mallows model. Let $\Phi_\sigma^{-1}(r) = (j_1, \dots, j_m)$ be the insertion ranks. If we multiply the factors ϕ^{i-j_i} across $i \leq m$ this gives $\phi^{\sum_{i=1}^m i - j_i} = \phi^{d(r, \sigma)}$ by Proposition 5. This term $\phi^{d(r, \sigma)}$ is exactly the proportional probability of r in Mallows. The denominator of $\prod_{i=1}^m p_{ij_i}$ is $(1 + \phi)(1 + \phi + \phi^2) \dots (1 + \phi + \dots + \phi^{m-1})$ regardless of r —this is exactly the normalizing constant in Mallows model. Interestingly, this gives an alternate proof of the normalization constant in the Mallows model. ■

Thus RIM offers a simple, useful way to sample rankings from the Mallows model while maintaining consistent partial rankings at each stage. In contrast to the rejection sampling approach of Condorcet/Mallows, RIM can be much more effective since it does not require the rejection of intransitive triads (which may occur with high probability if ϕ is large). We summarize the RIM approach from Mallows model:

RIM Sampling of Mallows

1. Let $\sigma = \sigma_1 \cdots \sigma_m$ be the reference ranking and ϕ the dispersion.
2. Start with an empty ranking r .
3. For $i = 1..m$:
 - Insert σ_i into r at rank position $j \leq i$ with probability $\phi^{i-j}/(1 + \phi + \cdots + \phi^{i-1})$.

RIM has worst-case quadratic running time (required number of draws from a Bernoulli distribution) when sampling from a Mallows model (this can be explained in much the same way as the complexity of insertion sort). However, the average-case time complexity can be much smaller, since insertions at each stage of the algorithm are likely to occur near the bottom of the partial ranking.

Proposition 7 *The expected time complexity of repeated insertion sampling for a Mallows model (σ, ϕ) is*

$$O\left(\min\left\{\frac{m(1 + \phi^{m+1})}{1 - \phi} - \frac{\phi(1 - \phi^m)}{(1 - \phi)^2}, m^2\right\}\right).$$

Proof Suppose we have $O(1)$ access to biased coin flips. The implementation will be as follows. Place σ_1 in the first rank. Then loop for $i = 2$ to m . Let $p_{ij} = \phi^{i-j} / \sum_{j'=0}^{i-1} \phi^{j'}$. Sample a rank position j to insert σ_i : start with $j = i$, flip a coin with probability p_{ij} , if success insert at rank j . Otherwise decrease j by 1, flip a coin with probability $p_{ij}/(1 - \sum_{j'>j} p_{ij'})$, if success, insert at rank j , otherwise decrease j by 1 and repeat this process until $j = 1$. By the chain rule, the probability of insertion at rank j is exactly what Mallows model requires. For each σ_i , when the sampled insertion rank position is j , it would require at most $i - j + 1$ coin flips. The expected running time, i.e., total number of coin flips, if $\phi < 1$, is proportional to

$$\begin{aligned} \sum_{i=1}^m \frac{\sum_{j=0}^{i-1} (j+1)\phi^j}{\sum_{j=0}^{i-1} \phi^j} &= \sum_{i=1}^m \frac{1}{1 - \phi} - i\phi^i \\ &\leq \frac{m(1 + \phi^{m+1})}{1 - \phi} - \frac{\phi(1 - \phi^m)}{(1 - \phi)^2}. \end{aligned}$$

This means one can effectively sample in linear time if ϕ is not too close to 1. If $\phi = 1$, the expected running time is $O(m^2)$. ■

Sampling with Weighted Kendall-tau. To illustrate the flexibility of RIM, we show it can be used to sample from a Mallows model using a weighted Kendall-tau distance. For two rankings r and σ and insertion vector $\mathbf{j} = (j_1, \dots, j_m)$ such that $\Phi_\sigma(\mathbf{j}) = r$, one can define a *weighted Kendall-tau distance* (Shieh, 1998) with respect to positive weights $\mathbf{w} = (w_1, \dots, w_m)$ as follows

$$d_{\mathbf{w}}(r, \sigma) = \sum_{i=1}^m w_i(i - j_i).$$

Recall that by Proposition 5, if $\mathbf{w} = \mathbf{1}$, then $d_{\mathbf{w}}$ is the standard Kendall-tau distance. Otherwise, this weighted Kendall-tau is sensitive to the pairwise misorderings of top-ranked alternatives in σ .

One can sample from a Mallows model defined by $P_{\mathbf{w}}(r) \propto e^{-d_{\mathbf{w}}(r,\sigma)}$ using RIM as follows. Let $\phi_i = e^{-w_i}$ for $i \leq m$. If we define the insertion probability of σ_i at position $j_i \leq i$ to be $\phi_i^{i-j_i} / (1 + \phi_i + \dots + \phi_i^{i-1})$, then the probability of generating r is proportional to $e^{\sum_{i=1}^m (i-j_i) \ln \phi_i} = e^{-d_{\mathbf{w}}(r,\sigma)}$.

2.3 A Mallows Mixture Model for Incomplete Preferences

While distributions such as Mallows or its mixture formulation (Murphy and Martin, 2003) give rise to complete rankings, there is relatively little work on generative models for partial rankings, and in particular, models that generate arbitrary (consistent) sets of pairwise comparisons. We introduce such a generative model in this section upon which to base our subsequent learning and inference procedures given such pairwise evidence.

A *Mallows mixture distribution* with K components is parameterized by mixing proportions $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$, reference rankings $\boldsymbol{\sigma} = (\sigma^{(1)}, \dots, \sigma^{(K)})$, and dispersion parameters $\boldsymbol{\phi} = (\phi_1, \dots, \phi_K)$. Rankings are generated randomly by selecting one of the K components according to the multinomial distribution with parameters $\boldsymbol{\pi}$. We sometimes represent this with a unit *component indicator* vector $\mathbf{z} = (z_1, \dots, z_K) \in \{0, 1\}^K$ in which the only entry of \mathbf{z} set to 1 is that of the selected component. If $z_k = 1$, then ranking r is drawn from the Mallows distribution with parameters $\sigma^{(k)}, \phi_k$.

In our model for partial preferences, we assume that each agent ℓ possesses a latent ranking r , where r is drawn from a mixture of Mallows distributions. We obtain the set of pairwise comparisons for ℓ by assuming a single additional parameter α which generates random pairs of alternatives. Intuitively, this reflects a process in which, given ℓ 's latent ranking r , each pair of alternatives is selected independently with probability α , and ℓ 's preference for that pair, as dictated by r , is revealed. That is,

$$P(v | r, \alpha) = \begin{cases} \alpha^{|v|} (1 - \alpha)^{\binom{m}{2} - |v|} & \text{if } r \in \Omega(v), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

This model reflects the relatively straightforward *missing at random assumption* (Ghahramani and Jordan, 1995), in which there is no correlation among those pairwise preferences that are missing/observed, nor any between observed pairs and the underlying ranking (e.g., the positions of the observed pairs). The missing at random assumption is not always realistic (Marlin and Zemel, 2007). We also note that this model assumes a single global parameter α that indicates the expected degree of completeness of each agent ℓ 's partial preferences. Allowing agent-specific completeness parameters α_ℓ and moving beyond “missing at random” are important directions. However, this model serves as a reasonable starting point for investigation.

Figure 1 illustrates a graphical model for the entire process. The resulting joint distribution is

$$P(v, r, \mathbf{z} | \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}, \alpha) = P(v | r, \alpha) P(r | \mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\phi}) P(\mathbf{z} | \boldsymbol{\pi}). \tag{10}$$

In our basic inference and learning problem, we take the observed data to be a preference profile $V = (v_1, \dots, v_n)$ of n agents, and we let $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$ denote the corresponding latent component memberships (i.e., \mathbf{z}_i indicates the mixture component where v_i is generated from).

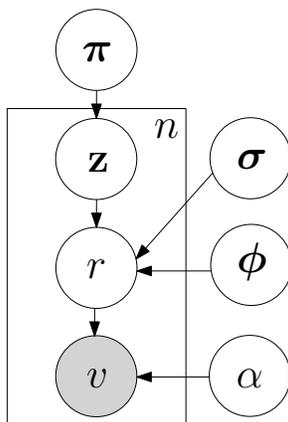


Figure 1: The generative model of incomplete preferences. Observed data v , a set of pairwise comparisons, is shaded.

2.4 Related Work

There is a large literature on ranking in the machine learning, statistics, economics, and theory of computation communities. It includes a variety of approaches, evaluation criteria, heuristics and applications, driven by several distinct motivations. In this section we briefly review two somewhat distinct lines of research.

The first body of work is that on *rank aggregation*. Roughly speaking, the aim is to find the best *objective ranking* given complete or partial observations generated by some noisy process involving the (latent) objective ranking. For example, such a ranking may be a ranking of web pages expressing a typical user’s (relative) degree of satisfaction with the pages. Observed information may consist of feedback, in the form of expert ratings or user preferences expressed implicitly via web page clicks on a search results page. In other applications, observed data may include partial rankings (e.g., in political elections), or pairwise comparisons (e.g., in sports leagues). Given such feedback, the ranking system will aggregate and optimize some objective function that attempts to capture user or population satisfaction such as NDCG—common in the IR field—(Burges et al., 2005; Volkovs and Zemel, 2009), misordered pairs (Cohen et al., 1999; Freund et al., 2003; Joachims, 2002; R. Herbrich and Obermayer, 2000), binary relevance (Agarwal and Roth, 2005; Rudin, 2009), and objectives from social choice theory (e.g., Kemeny, Borda rankings). For example, in machine learning, the area of *learning to rank (LETOR)* has been a topic of much research since the late 1990s, starting with the work of Cohen et al. (1999). Research into ranking systems often seeks strong *generalization* capabilities, in the sense that it can produce an objective ranking given a previously unencountered ranking problem using new attributes (e.g., rank web pages given a new search query). Much of this research has indeed been focused on web ranking applications (e.g., the Yahoo! Learning to Rank Challenge; see Burges, 2010). More recently, Busa-Fekete et al. (2014) have developed active learning algorithms for inferring certain distributional properties of the Mallows model.

There are also communities in statistics and computational social choice that are concerned with estimating the maximum likelihood ranking under some distributional assumptions. Often such models—for example, the Mallows and Plackett-Luce models discussed above—assume a central, modal or reference objective ranking at which the distribution is peaked. A fundamental problem is estimation of this objective ranking from a collection of ordinal preference data. For example, the Kemeny ranking can be interpreted as a maximum likelihood estimate of the modal ranking in a Mallows model (Young, 1995). Other such interpretations of common rank aggregation rules also exist (Conitzer and Sandholm, 2005; Conitzer et al., 2009).

The above perspective, that of computing an *objective* ranking, applies to many situations (e.g., one would expect the ranking of web pages for a search query in “norovirus symptoms” to be objectively stable, since users will largely agree the informativeness of retrieved web pages). However, in many settings this is entirely inappropriate. When a group of individuals plans an activity together, such as going to a restaurant for dinner, the ranking of restaurants should clearly depend on the personal tastes and preferences of the individuals involved. In such cases, a distribution over a population’s *subjective preferences* better reflects reality. A second, growing, body of work aims to assess (individual or aggregate/group) rankings of options, or decisions, by explicitly using, modelling or reasoning about the diversity of user preferences. This is a more general problem than that of objective rank aggregation. For example, the Netflix collaborative filtering competition has initiated much research on predicting a user’s movie ratings given the ratings for other movies, including their own and those of other users. Other relevant research on such ranking work includes *label ranking* (Hüllermeier et al., 2008), which seeks to aggregate sparse preference data of “similar users” into personalized preferences.

In recent years there has been growing interest in applying probabilistic models of preferences from statistics, psychometrics, and econometrics to model a population’s preferences. This is the context in which our work is situated. We focus on learning such preference distributions, including multimodal distributions over preferences where each mode (cluster) corresponds to a “sub-type” within the population. Much recent research has focused on using the single-peaked Mallows model as a basis for multimodal mixture distributions. One of the first papers to propose an algorithm for learning Mallows mixtures is that of Murphy and Martin (2003). Their method assumes that training data takes the form of *complete* preference rankings (individual preferences), and has a running time that is factorial in the number of alternatives. Busse et al. (2007) develop a tractable EM algorithm for Mallows mixtures where preferences are restricted to be of the top- t type. A recent extension by Meila and Chen (2010) of Mallows mixtures allows for a Bayesian treatment in choosing the number of components using Dirichlet process mixtures, and offers experiments on considerably larger data sets. Recent work has also studied fitting temporal mixture models (a variation on the Bradley-Terry model) using EM (Francis et al., 2014).

Aside from mixture models, Lebanon and Mao (2008) propose a non-parametric kernel density estimator for rankings, which places a “smooth Mallows bump” on each training preference. They derive an efficiently computable, closed-form formula for the evaluation of the estimator. However, they restrict their training data to partitioned preferences (see above), a more general concept than top- t rankings, but significantly less expressive than arbitrary pairwise comparisons. In contrast to our work, they do not address how to learn

the kernel bandwidth parameter (see Section 6 for further discussion). There has been recent work on sampling algorithms for rankings that shares some similarities with the GRIM algorithm we develop here. This includes a sampling algorithm based on a generalization of the Plackett-Luce model (Volkovs and Zemel, 2012), inspired by bipartite matching problems that occur in certain application domains. Biernacki and Jacques (2013) propose a noisy insertion-sort model of rankings and develop EM algorithms for estimating its parameters. This is related to RIM but with some minor differences. However, none of this work addresses the question of sampling from a posterior distribution given partial preferences as evidence.

Apart from the the Mallows model, the Plackett-Luce model has also been popular as a representation of preferences. Recent work on learning and inference with this model includes: an approach to Bayesian inference of the modal ranking (Guiver and Snelson, 2009), but where training preferences are limited to ranking of all of alternatives in some subset of alternatives; and a method for learning a mixture model given top- k preferences (Gormley and Murphy, 2008), with application to political voting data.

Huang and Guestrin (2009) develop the *riffle independence model*, which partitions a set of alternatives into two sets: a ranking of each set is generated stochastically (and independently); then a stochastic process is used to interleave or “riffle” the two resulting rankings to produce a combined ranking. The model can applied hierarchically, with the same process used to generate the required subrankings. Huang et al. (2012) show that inference in this model is tractable for certain classes of observations. Of particular note is that fact that conditioning on partitioned preferences (which they term “partial ranking observations”) can be accomplished efficiently. Interestingly, Mallows models can be represented using the riffle independence model.

3. Generalized Repeated Insertion Model

Our ultimate goal is to support effective learning and inference with Mallows models (and by extension, Mallows mixtures) given observed data or evidence in the form of partial preference profiles consisting of arbitrary pairwise comparisons. Sampling is, of course, an important aspect of this. The rejection sampling models discussed above can obviously be extended to accommodate pairwise observations, but are likely to be extremely inefficient. By contrast, while RIM provides a powerful tool for sampling from Mallows models (and mixtures), it samples unconditionally, without allowing for (direct) conditioning on evidence. In this section, we describe and analyze a generalized version of the RIM technique that permits conditioning at each insertion step. In fact, our *generalized repeated insertion model (GRIM)* can be used to sample from *arbitrary* rank distributions. We begin in Section 3.1 by describing GRIM in this general, abstract fashion. The primary focus of our theoretical and computational analysis in Section 3.2, however, will be on its use for Mallows distributions.

3.1 Sampling from Arbitrary Ranking Distributions

We first present the *generalized repeated insertion model (GRIM)* abstractly as a means of sampling from *any* distribution over rankings. GRIM is based on a relatively simple insight, namely, that the chain rule allows us to represent any distribution over rankings in a concise

way, as long as we admit *dependencies* in our insertion probabilities. Specifically, we allow the insertion probabilities for any alternative σ_i in the reference ranking to be *conditioned on the ordering* of the previously inserted alternatives $(\sigma_1, \dots, \sigma_{i-1})$.

Let Q be any distribution over rankings and σ an (arbitrary) reference ranking. Recall that we can (uniquely) represent any ranking $r \in \Omega$ using σ and an insertion vector $\mathbf{j}^r = (j_1^r, \dots, j_m^r) \in I$, where $r = \Phi_\sigma(\mathbf{j}^r)$. Thus Q can be represented by a distribution Q' over the space I of insertion vectors, i.e., $Q'(\mathbf{j}^r) = Q(r)$. Similarly, for $k < m$, any partial ranking $r[k] = (r_1, \dots, r_k)$ of the alternatives $\{\sigma_1, \dots, \sigma_k\}$, can be represented by a partial insertion vector $\mathbf{j}[k] = (j_1^r, \dots, j_k^r)$. Letting

$$Q(r[k]) = \sum \{Q(r) : r_1 \succ r_2 \succ \dots \succ r_k\} \quad \text{and} \quad Q'(\mathbf{j}[k]) = \sum \{Q'(\mathbf{j}') : \mathbf{j}'[k] = \mathbf{j}[k]\},$$

we have $Q'(\mathbf{j}[k]) = Q(r[k])$. We define *conditional insertion probabilities*:

$$p_{ij|\mathbf{j}[i-1]} = Q'(j_i = j | \mathbf{j}[i-1]). \tag{11}$$

This denotes the probability with which the i th alternative σ_i in the reference ranking is inserted at position $j \leq i$, *conditioned on the specific insertions* $(j_1^r, \dots, j_{i-1}^r)$ of all previous alternatives. By the chain rule, we have

$$Q'(\mathbf{j}) = Q'(j_m|\mathbf{j}[m-1])Q'(j_{m-1}|\mathbf{j}[m-2]) \dots Q'(\mathbf{j}[1]).$$

Suppose we apply RIM with conditional insertion probabilities $p_{ij|\mathbf{j}[i-1]}$ defined above; that is, we draw random insertion vectors \mathbf{j} by sampling j_1 through j_m , in turn, but with each conditioned on the previously sampled components. The chain rule ensures that the resulting insertion vector is sampled from the distribution Q' . Hence the induced distribution over rankings $r = \Phi_\sigma(\mathbf{j})$ is Q . We call the aforementioned procedure the *generalized repeated insertion model (GRIM)*. Based on the arguments above, we have:

Theorem 8 *Let Q be any ranking distribution and σ a reference ranking. For any $r \in \Omega$, with insertion vector \mathbf{j}^r (i.e., $r = \Phi_\sigma(\mathbf{j}^r)$), GRIM, using the insertion probabilities in Equation 11, generates insertion vector \mathbf{j}^r with probability $Q'(\mathbf{j}^r) = Q(r)$.*

For instance, GRIM can be used to sample from a (conditional) Mallows model given evidence in the form of pairwise comparisons, as shown in the following example.

Example 1 *We illustrate GRIM using a simple example, sampling from a (conditional) Mallows model over $A = \{a, b, c\}$, with dispersion ϕ , given evidence $v = \{a \succ c\}$. The following table describes the steps in the process:*

| Insert a, b | | Insert c given ab | | Insert c given ba | |
|-------------|----------------------------------|-------------------|----------------------------------|-------------------|-----------------|
| r | Insertion Prob. | r | Insertion Prob. | r | Insertion Prob. |
| a | $P(j_a=1) = 1$ | cab | $P(j_c=1) = 0$ | cba | $P(j_c=1) = 0$ |
| ab | $P(j_b=1) = \frac{1}{1+\phi}$ | acb | $P(j_c=2) = \frac{\phi}{1+\phi}$ | bca | $P(j_c=2) = 0$ |
| ba | $P(j_b=2) = \frac{\phi}{1+\phi}$ | abc | $P(j_c=3) = \frac{1}{1+\phi}$ | bac | $P(j_c=3) = 1$ |

The resulting ranking distribution Q is given by the product of the conditional insertion probabilities: $Q(abc) = 1/(1 + \phi)^2$; $Q(acb) = \phi/(1 + \phi)^2$; and $Q(bac) = \phi/(1 + \phi)$. As required, $Q(r) = 0$ iff r is inconsistent with evidence v .

3.2 Sampling from Mallows Posteriors

We now develop and analyze several techniques for sampling from (mixtures of) Mallows models given partial preference profiles as evidence. We use the term *Mallows posterior* to refer to the conditional distribution that arises from incorporating evidence—in the form of a set of pairwise comparisons—into a known Mallows model. This is the primary inference task facing a system making predictions about a specific user’s preferences given pairwise evidence from that user, assuming a reasonably stable population model. This stands in contrast to the more general problem of learning the parameters of a Mallows model (a problem we address in Section 4).

3.2.1 INTRACTABILITY OF SAMPLING

One key difficulty with enabling inference conditioned on pairwise comparisons is the intractability of the posterior. In the above model (Equation 10), where agent ℓ ’s incomplete preference v_ℓ is observed, it is intractable to work with the posterior $P(r, \mathbf{z} | v_\ell, \boldsymbol{\pi}, \boldsymbol{\sigma}, \phi, \alpha)$ even when the mixture model has a single component, a fact we prove below. One typical approach is to rely on sampling to estimate the posterior. To this end, we develop a polynomial-time posterior sampling algorithm based on GRIM, but relying on approximation of the relevant conditional insertion probabilities.

While GRIM allows sampling from arbitrary distributions over rankings, as presented above it is largely a theoretical device, since it requires inference to compute the required conditional probabilities. Thus to use GRIM to sample from a Mallows posterior, given arbitrary pairwise comparisons v , we must first derive these required terms. The Mallows posterior is given by

$$P_v(r) = P(r | v) = \frac{\phi^{d(r,\sigma)}}{\sum_{r' \in \Omega(v)} \phi^{d(r',\sigma)}} \mathbf{1}[r \in \Omega(v)], \tag{12}$$

which requires summing over an intractable number of rankings to compute the normalization constant.

We could use RIM for rejection sampling: sample unconditional insertion ranks, and reject a ranking at any stage if it is inconsistent with v . However, this is impractical because of the high probability of rejection. One can also modify the pairwise comparison sampling model (see Section 2.2.2) to reject inconsistent pairwise comparisons. However, if $|v|$ is small relative to m , then for values of ϕ that are not too small, the probability of rejection is very high. For instance, if ϕ is close to 1, $m = 120$ and 30 alternatives appear in v , any three alternatives the probability of a cyclic triad for any triple (e.g., $a \succ b, b \succ c, c \succ a$) is $\approx 1/4$. The 90 alternatives unconstrained by v can be divided into 30 groups of 3 alternatives, hence the probability that a cycle occurs among at least one triad is at least $1 - (3/4)^{30} = 0.9998$. This is a lower bound on the probability of rejection, showing rejection sampling to be impractical in many settings.

The main obstacle to using GRIM for sampling is computation of the insertion probabilities of a specific alternatives given the inserted positions all previous alternatives, as given by Equation 11, when Q' (more precisely, the corresponding Q) is the Mallows posterior. This essentially involves computing a high-order marginal over rankings, and turns

out to be #P-hard, even with a uniform distribution over $\Omega(v)$. The following result on the complexity of counting linear extensions of a partial order will be useful below:

Theorem 9 (Brightwell and Winkler 1991) *Given a partial order v , computing the number of linear extensions of v , that is $|\Omega(v)|$, is #P-complete.*

To show that computing a function $f(x)$ is #P-hard for input x , it is sufficient to show that a #P-complete problem can be reduced to it in polynomial time.

Proposition 10 *Given v , a reference ordering σ , a partial ranking $r_1 \cdots r_{i-1}$ over $\sigma_1, \dots, \sigma_{i-1}$, and $j \leq i$, computing the probability of inserting σ_i at rank j with respect to the uniform Mallows posterior P (i.e., computing $P(r) \propto \mathbf{1}[r \in \Omega(v)]$) is #P-hard.*

Proof We reduce the problem of counting the number of linear extensions of incomplete preferences v , which is a #P-complete problem, to that of computing the desired insertion probabilities, showing the problem to be #P-hard. Given v , notice that any $r = r_1 \dots r_m \in \Omega(v)$ has a uniform posterior probability of $1/|\Omega(v)|$. Let $\Phi_\sigma^{-1}(r) = (j_1, \dots, j_m)$. Assume the existence of an algorithm f to compute the required insertion probabilities. We can use it to solve the counting problem as follows: we use f to compute $p_{ij_i} = \Pr(\text{insert } \sigma_i \text{ at rank } j_i \mid r_{\{\sigma_1, \dots, \sigma_{i-1}\}})$ with partial order v for each $i \in \{2, \dots, m\}$ (i.e., $m - 1$ applications of f). By Theorem 8, we know the posterior probability of r is $1/|\Omega(v)| = \prod_i p_{ij_i}$; thus we can compute $|\Omega(v)|$ by inverting the product of the insertion probabilities. Note that this reduction can be computed in polynomial time: we can construct any $r \in \Omega(v)$ by using a topological sort algorithm, and we require only $m - 1$ calls to the algorithm insertion algorithm f . ■

This result shows that it is hard to sample exactly in general, and suggests that computing the normalization constant in a Mallows posterior is difficult. This would also imply a computational complexity obstacle in the work on non-parametric estimators with a Mallows kernel (Lebanon and Mao, 2008) for an *arbitrary* set of pairwise comparisons. Nevertheless we develop an *approximate sampler* AMP that is computationally very efficient. While its approximation quality can be quite poor in the worst case, we see below that, empirically, it produces excellent posterior approximations. We also derive bounds that delineate circumstances under which it will provide approximations with low error.

3.2.2 AMP: AN APPROXIMATE SAMPLER

AMP is based on the same intuitions as those illustrated in Example 1, where instead of computing the correct insertion probabilities, we use the (unconditional) insertion probabilities used by RIM, but subject to constraints imposed by v . First, we compute the transitive closure $\text{tc}(v)$ of v . Then we use a modified repeated insertion procedure where at each step, the alternative being inserted can only be placed in positions that do not contradict $\text{tc}(v)$. We can show that the valid insertion positions for any alternative, given v , form a *contiguous region* of the ranking (see Figure 2 for an illustration).

Proposition 11 *Given partial preference v , let the insertion of $i-1$ alternatives $\sigma_1, \dots, \sigma_{i-1}$ induce a ranking $r_1 \cdots r_{i-1}$ that is consistent with $\text{tc}(v)$. Let $L_i = \{i' < i \mid r_{i'} \succ_{\text{tc}(v)} \sigma_i\}$ and*

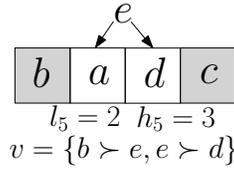


Figure 2: Valid insertion ranks for e are $\{l_5, \dots, h_5\} = \{2, 3\}$ given previous insertions and constraints v .

Algorithm 1 AMP Approximate Mallows Posterior

Input: v, σ, ϕ

- 1: $r \leftarrow \sigma_1$
- 2: **for** $i = 2..m$ **do**
- 3: Calculate l_i and h_i from Equations 13 and 14.
- 4: Insert σ_i in r at rank $j \in \{l_i, \dots, h_i\}$ with probability $\frac{\phi^{i-j}}{\sum_{l_i \leq j' \leq h_i} \phi^{i-j'}}$.
- 5: **end for**

Output: r

$H_i = \{i' < i \mid r_{i'} \prec_{\text{tc}(v)} \sigma_i\}$. Then inserting σ_i at rank j is consistent with $\text{tc}(v)$ if and only if $l_i \leq j \leq h_i$, where

$$l_i = \begin{cases} 1 & \text{if } L_i = \emptyset \\ \max(i' \in L_i) + 1 & \text{otherwise,} \end{cases} \tag{13}$$

$$h_i = \begin{cases} i & \text{if } H_i = \emptyset \\ \min(i' \in H_i) & \text{otherwise.} \end{cases} \tag{14}$$

Proof Inserting σ_i at any rank position less than l_i is impossible since either $l_i = 1$ (we can't insert in rank 0) or σ_i lies above r_{l_i} , which contradicts the requirement imposed by $\text{tc}(v)$ that r_{l_i} must be ranked higher. A similar argument can be made for inserting in rank below h_i since r_{h_i} needs to be below σ_i . Finally, inserting into any rank in $\{l_i, \dots, h_i\}$ does not violate $\text{tc}(v)$ since the alternative will be inserted below *all* alternatives that must precede it in $\text{tc}(v)$ and *all* alternatives that must succeed it. ■

Proposition 11 immediately suggests an implementation of the GRIM algorithm, AMP, for approximate sampling of the Mallows posterior—AMP is outlined in Algorithm 1. It first initializes ranking r with σ_1 at rank 1. Then for each $i = 2 \dots m$, it computes l_i, h_i and inserts σ_i at rank $j \in \{l_i, \dots, h_i\}$ with probability proportional to ϕ^{i-j} . Note that $\text{tc}(v)$, which is required as part of the algorithm, can be computed via a modified depth-first search. AMP induces a sampling distribution \hat{P}_v that does not match the posterior P_v exactly: indeed the KL-divergence between the two can be severe, as the following example shows.

Example 2 Let $A = \{a_1, \dots, a_m\}$ and $v = a_2 \succ a_3 \succ \dots \succ a_m$. Let P be the uniform Mallows prior ($\phi = 1$) with $\sigma = a_1 \dots a_m$. There are m rankings in $\Omega(v)$, one ranking

r_i for each placement of a_1 into rank position $1 \leq i \leq m$. That is, $r_1 = a_1 a_2 \cdots a_m$ and $r_i = a_2 \cdots a_i a_1 a_{i+1} \cdots a_m$ for $i \geq 2$. The true Mallows posterior P_v is uniform over $\Omega(v)$. But AMP induces an approximation with $\hat{P}_v(r_i) = 2^{-i}$ for $i \leq m - 1$ and $\hat{P}_v(r_m) = 2^{-m-1}$. To see this, note that to construct r_i , AMP would need to insert alternatives a_2, \dots, a_i successively, each with probability $1/2$, above a_1 . Then a_{i+1} must be inserted below a_1 with probability $1/2$, and finally the remaining alternatives a_{i+2}, \dots, a_m can only be inserted at the bottom (with probability 1). Hence, the KL-divergence between P_v and \hat{P}_v is

$$\begin{aligned} KL(P_v || \hat{P}_v) &= \sum_{i=1}^m P_v(r_i) \log_2 \left(\frac{P_v(r_i)}{\hat{P}_v(r_i)} \right) \\ &= \left[\sum_{i=1}^{m-1} \frac{1}{m} \log_2 \frac{1/m}{2^{-i}} \right] + \frac{1}{m} \log_2 \frac{1/m}{2^{-m-1}} \\ &= 1 - \frac{1}{m} + \frac{m-1}{2} - \log_2 m . \end{aligned}$$

3.2.3 STATISTICAL PROPERTIES OF AMP

Example 2 shows that AMP may provide poor approximations in the worst case; however we will see below (Section 5) that it performs very well in practice. We can also prove interesting properties, and provide theoretical guarantees of exact sampling in important special cases.

We first observe that AMP always produces a valid ranking; in other words, valid insertion positions always exist given any consistent v .

Proposition 12 *For all $i \geq 2$ and all rankings of alternatives $\sigma_1, \dots, \sigma_{i-1}$ that is consistent with v , we have that $l_i \leq h_i$, where l_i and h_i are defined in Equation 13 and 14, respectively. That is, AMP always has a position at which to insert alternative σ_i .*

Proof Let r be a ranking of $\sigma_1, \dots, \sigma_{i-1}$ consistent with v . Let x be the lowest ranking alternative in r such that $x \succ_{\text{tc}(v)} \sigma_i$ and y the highest-ranked alternative in r with $y \prec_{\text{tc}(v)} \sigma_i$. By transitivity, $x \succ_{\text{tc}(v)} y$. Now if $h_i < l_i$ (as defined in terms of r) this implies $y \succ_r x$, but this contradicts the assumption that r is consistent with v . ■

Furthermore, the approximate posterior has the same support as the true posterior:

Proposition 13 *The support of the distribution over rankings as defined by AMP is equal to $\Omega(v)$ which is equal to the support of the Mallows posterior as given in Equation 12.*

Proof By Proposition 11, the algorithm never violates the constraints in $\text{tc}(v)$, and by Proposition 12, it will always have at least one valid insertion position. Hence the algorithm always outputs a ranking consistent with v . Now, let $r \in \Omega(v)$ and $\Phi_\sigma^{-1}(r) = (j_1, \dots, j_m)$ be the its corresponding insertion vector. We show that for all $i \leq m$, $j_i \in \{l_i, \dots, h_i\}$. If this is not true, then there exists a smallest $i' \leq m$ such that $j_{i'} \notin \{l_{i'}, \dots, h_{i'}\}$ (note $i' \geq 2$ since the first alternative is always inserted at the first position). However, Proposition 11 asserts that this insertion rank would lead to a ranking inconsistent with v —so this is not possible. Since AMP places positive probability on any insertion position in $\{l_i, \dots, h_i\}$ then r has positive probability under AMP. ■

Proposition 14 For any $r \in \Omega(v)$, the probability AMP will output r is

$$\hat{P}_v(r) = \frac{\phi^{d(r,\sigma)}}{\prod_{i=1}^m (\phi^{i-h_i} + \phi^{i-h_i+1} + \dots + \phi^{i-l_i})}. \tag{15}$$

Proof Let $\Phi_\sigma^{-1}(r) = (j_1, \dots, j_m)$ be the insertion ranks. We have already established in Proposition 13 that AMP puts positive probability on these valid insertion ranks. In fact the probability of r under the algorithm (see Algorithm 1) is

$$\begin{aligned} \prod_{i=1}^m \frac{\phi^{i-j_i}}{(\phi^{i-l_i} + \phi^{i-l_i-1} + \dots + \phi^{i-h_i})} &= \frac{\phi^{\sum_{i=1}^m i-j_i}}{\prod_{i=1}^m (\phi^{i-l_i} + \phi^{i-l_i-1} + \dots + \phi^{i-h_i})} \\ &= \frac{\phi^{d(r,\sigma)}}{\prod_{i=1}^m (\phi^{i-l_i} + \phi^{i-l_i-1} + \dots + \phi^{i-h_i})}, \end{aligned}$$

where the last equality comes from Proposition 5. ■

Using this result we can show that if v lies in the class of partitioned preferences, AMP’s induced distribution is exactly the Mallows posterior:

Proposition 15 (Lebanon and Mao 2008) Let σ be a reference ranking. Let v be a partitioned preference (see Definition 1) with partition A_1, \dots, A_q of A . Let $\delta = |\{(x, y) | y \succ_\sigma x, x \in A_i, y \in A_j, i, j \in [q], i < j\}|$, which is the number of pairs of alternatives, that span different subsets of the partition, that are misordered with respect to σ . Then

$$\delta = \sum_{i=1}^{q-1} \sum_{x \in A_i} \sum_{j=i+1}^q \sum_{y \in A_j} \mathbf{1}[y \succ_\sigma x], \tag{16}$$

$$\sum_{r \in \Omega(v)} \phi^{d(r,\sigma)} = \phi^\delta \prod_{i=1}^q \prod_{j=1}^{|A_i|} (1 + \phi + \phi^2 + \dots + \phi^{j-1}). \tag{17}$$

Notice that Equation 17 represents the normalization constant in Mallows posterior. The intuition underlying Equation 17 is that, for any $r \in \Omega(v)$, the misorderings contributed by alternatives that span two subsets, as given by δ , are the same (hence the leading factor) whereas within a subset A_i alternatives can be ordered arbitrarily (hence the product of normalization constants for $|A_i|$).

Proposition 16 Given a partitioned preference v , the distribution induced by AMP, \hat{P}_v , is equal to the true Mallows posterior P_v .

Proof Since the numerator in Equation 15 (which denotes the probability that AMP outputs r) is the same as the proportional probability of the Mallows posterior, it is sufficient to show that the denominator in Equation 15 equals the Mallows posterior normalization constant given by Equation 17. Suppose $\sigma = \sigma_1 \dots \sigma_m$. Let v be a partitioned preference A_1, \dots, A_q . Consider alternatives in A_i such that $\sigma|_{A_i} = \sigma_{t_1} \sigma_{t_2} \dots \sigma_{t_{|A_i|}}$ (i.e., the ranking of alternatives in A_i according to σ). For any $k \in \{1, \dots, |A_i|\}$, suppose alternatives $A' = \{\sigma_1, \dots, \sigma_{t_{k-1}}\}$ are inserted. The structure of the resulting ranking is as follows: the alternatives $(A_1 \cup A_2 \cup \dots \cup A_{i-1}) \cap A'$ must lie at the top of the ranking; the alternatives

$A_i \cap A' = \{\sigma_{t_1}, \dots, \sigma_{t_{k-1}}\}$ are in the middle; and $B_{t_k} = (A_{i+1} \cup \dots \cup A_q) \cap A'$ are at bottom. When inserting σ_{t_k} at rank j , we have $j \in \{l_{t_k}, \dots, h_{t_k}\}$, where $h_{t_k} = t_k - |B_{t_k}|$ and $l_{t_k} = h_{t_k} - |A_i \cap A'| = t_k - (k-1) - |B_{t_k}|$. Hence σ_{t_k} is inserted at rank j with probability

$$\frac{\phi^{t_k-j}}{\phi^{t_k-h_{t_k}} + \dots + \phi^{t_k-l_{t_k}}} = \frac{\phi^{t_k-j}}{\phi^{|B_{t_k}|} + \dots + \phi^{k-1+|B_{t_k}|}}.$$

The denominator can be written $\phi^{|B_{t_k}|}(1 + \dots + \phi^{k-1})$. Observe that B_{t_k} consists of all alternatives from A' that are above σ_{t_k} in σ , but are below it in v (since all such alternatives belong to $A_{i+1} \cup \dots \cup A_q$). So $\sum_{k=1}^{|A_i|} |B_{t_k}|$ is the total number of pairs (x, y) , where $x \in A_i$ and $y \in A_{i+1} \cup \dots \cup A_q$, that are misordered with respect to σ . Thus inserting alternatives in A_i contributes a factor of

$$\prod_{k=1}^{|A_i|} \phi^{|B_{t_k}|} (1 + \dots + \phi^{k-1}) = \phi^{\sum_{x \in A_i} \sum_{j=i+1}^q \sum_{y \in A_j} \mathbf{1}[y >_{\sigma} x]} \prod_{k=1}^{|A_i|} (1 + \dots + \phi^{k-1})$$

to the denominator in Equation 15. Once all alternatives have been inserted, the denominator becomes

$$\phi^{\sum_{i=1}^q \sum_{x \in A_i} \sum_{j=i+1}^q \sum_{y \in A_j} \mathbf{1}[y >_{\sigma} x]} \prod_{i=1}^q \prod_{k=1}^{|A_i|} (1 + \dots + \phi^{k-1}).$$

This is exactly the Mallows posterior normalization constant in Equation 17. ■

As a consequence, AMP provides exact sampling in the case of partitioned preferences,

In general, this is not the case with arbitrary partial preferences (pairwise comparisons). We now derive bounds on the relative error of AMP's posterior, bounding the ratio between the sample probability of an arbitrary ranking r for AMP and the true posterior probability. The main technical challenge is deriving a bound on the Mallows posterior normalization constant. We can obtain an upper bound by exploiting the pairwise comparison interpretation of Mallows model (see Section 2.2.2).

Theorem 17 (Upper Bound on Normalization Constant) *Let σ be a reference ranking, $\phi \in (0, 1]$ and v a preference. The Mallows posterior normalization constant is upper bounded by*

$$\sum_{r \in \Omega(v)} \phi^{d(r, \sigma)} \leq \phi^{d(v, \sigma)} (1 + \phi)^{\binom{m}{2} - d(v, \sigma) - s(v, \sigma)}. \tag{18}$$

Proof The LHS of Equation 18 can be written in terms Equation 4, by setting $\phi = p/(1-p)$ (see Section 2.2.2 for derivations of the pairwise comparison interpretation of Mallows) as follows:

$$\begin{aligned} \sum_{r \in \Omega(v)} \phi^{d(r, \sigma)} &= Z \cdot \sum_{r \in \Omega(v)} P(r | \sigma, p) \\ &= Z \cdot \frac{1}{Z'} \sum_{r \in \Omega(v)} \prod_{\{x, y\} \subseteq A} \begin{cases} p & \text{if } r \text{ and } \sigma \text{ disagree on } x, y \\ 1-p & \text{otherwise,} \end{cases} \end{aligned} \tag{19}$$

where $p = \phi/(1 + \phi)$, Z' is given by Equation 6 and Z is given by Equation 3, thus the constant in front simplifies to $1/(1-p)^{\binom{m}{2}}$. Since r must be consistent with v , if x and y are comparable under v , then r must agree with v on (x, y) , i.e., if $x \succ_{\text{tc}(v)} y$ then $x \succ_r y$. So

$$P(r|\sigma, p) = \frac{1}{Z'} p^{d(v,\sigma)} (1-p)^{s(v,\sigma)} \prod_{\{x,y\} \notin \text{tc}(v)} \begin{cases} p & \text{if } r \text{ and } \sigma \text{ disagree on } x, y \\ 1-p & \text{otherwise.} \end{cases}$$

Hence, since $\Omega(v)$ is contained in the set of all *intransitive* relations on A that is consistent with comparisons in $\text{tc}(v)$, we must have (for $k = \binom{m}{2} - d(v, \sigma) - s(v, \sigma)$)

$$\begin{aligned} \sum_{r \in \Omega(v)} P(r|\sigma, p) &\leq \frac{1}{Z'} p^{d(v,\sigma)} (1-p)^{s(v,\sigma)} \sum_{z \in \{0,1\}^k} \prod_{i=1}^k p^{z_i} (1-p)^{1-z_i}, \\ &= \frac{1}{Z'} p^{d(v,\sigma)} (1-p)^{s(v,\sigma)}. \\ Z \cdot \sum_{r \in \Omega(v)} P(r|\sigma, p) &\leq \frac{1}{(1-p)^{\binom{m}{2}}} p^{d(v,\sigma)} (1-p)^{s(v,\sigma)}. \end{aligned} \tag{20}$$

Combining Equation 20 with Equation 19, and noting that $p = \phi/(1 + \phi)$, we obtain Equation 18. ■

Equation 18 tells us if $d(v, \sigma)$ increases (i.e., v increasingly disagrees with σ), then the first factor dominates and upper bound gets smaller—this reflects our natural intuitions since the set $\Omega(v)$ gets “further away” from reference ranking σ and hence its probability mass is small. We also see that if $|\text{tc}(v)|$ is small, then $d(v, \sigma) + s(v, \sigma)$ is small and the upper bound increases since the second factor dominates. This too makes sense because $\Omega(v)$ is large and has greater probability mass. If $s(v, \sigma)$ is large, more constraints are placed on v , hence $\Pr(\Omega(v))$ is smaller, and likewise the upper bound decreases. The following example illustrates that this bound may be quite loose in some cases, but tight in others.

Example 3 Consider again the partial ranking evidence from Example 2, where $v = a_2 \succ \dots \succ a_m$, the alternatives are $\{a_1, \dots, a_m\}$, and our reference ranking is $\sigma = a_1 a_2 \dots a_m$. Recall that there are m rankings in $\Omega(v)$, one ranking r_i for each placement of a_1 into rank position i . Now the term on the LHS of Equation 18, i.e., the true value of the normalization constant, is

$$\sum_{i=1}^m \phi^{d(r_i,\sigma)} = 1 + \phi + \phi^2 + \dots + \phi^m.$$

Note that $d(v, \sigma) = 0$ and $s(v, \sigma) = \binom{m-1}{2}$ since all pairwise comparisons in $\text{tc}(v)$ agree with σ . Thus, the term on the RHS of Equation 18, i.e., the upper bound is

$$\phi^0 (1 + \phi)^{\binom{m}{2} - 0 - \binom{m-1}{2}} = (1 + \phi)^{m-1}.$$

This upper bound on the normalization constant gets tight as $\phi \rightarrow 0$, but becomes exponentially loose in m as $\phi \rightarrow 1$.

Before we derive a lower bound, we introduce some notions from order theory.

Definition 18 Let v be a partial preference. An anti-chain of v is a subset X of A such that for every $x, y \in X$ they are incomparable under $\text{tc}(v)$. A maximum anti-chain is an anti-chain whose size is at least the size of any anti-chain. The width of v , $w(v)$ is the size of a maximum anti-chain of v .

Theorem 19 (Lower Bound on Normalization Constant) Let σ be a reference ranking, and $\phi \in (0, 1]$. Let X be a maximum anti-chain of v , $Y = \{a \in A \setminus X \mid \exists x \in X, a \succ_{\text{tc}(v)} x\}$ and $Z = A \setminus (X \cup Y)$. Let $\delta = |\{(x, y) \mid x \in X, y \in Y, x \succ_{\sigma} y\}| + |\{(y, z) \mid y \in Y, z \in Z, z \succ_{\sigma} y\}| + |\{(x, z) \mid x \in X, z \in Z, z \succ_{\sigma} x\}|$. Denote by $\text{tc}(v)|_Y$ and $\text{tc}(v)|_Z$ the transitive closure of v restricted to the subsets Y and Z , respectively. Also let $\Omega(\text{tc}(v)|_Y)$ denote those rankings over Y that are consistent with $\text{tc}(v)|_Y$, and similarly for $\Omega(\text{tc}(v)|_Z)$. We have

$$\sum_{r \in \Omega(v)} \phi^{d(r, \sigma)} \geq \phi^{\delta} \left[\sum_{r \in \Omega(\text{tc}(v)|_Y)} \phi^{d(r, \sigma|_Y)} \right] \left[\sum_{r \in \Omega(\text{tc}(v)|_Z)} \phi^{d(r, \sigma|_Z)} \right] \prod_{i=1}^{w(v)} \sum_{j=0}^{i-1} \phi^j. \quad (21)$$

Proof We first show that $Z' = \{a \in A \setminus X \mid \exists x \in X, x \succ_{\text{tc}(v)} a\} = Z$. If $a \in A \setminus X$ does not belong to Y then it must be comparable to at least one element in $x \in X$ otherwise we can add it to Y and obtain a larger anti-chain. Hence, since a is not in Y , then $x \succ_{\text{tc}(v)} a$. Also, note that if $a \in Y$ then $a \notin Z'$. This is because if a belonged to both Y and Z , then there exists $x_1, x_2 \in X$ such that $x_1 \succ_{\text{tc}(v)} a$ and $a \succ_{\text{tc}(v)} x_2$ this would mean $x_1 \succ_{\text{tc}(v)} x_2$ which contradicts the anti-chain property of X . For a particular alternative in X , alternatives in Y are either incomparable to it or must be preferred to it, similarly alternatives in Z are either incomparable or must be dis-preferred to it.

This also implies no alternative in Z can be preferred over alternatives in Y since if this were to happen, i.e., if $z \succ_{\text{tc}(v)} y$ where $z \in Z, y \in Y$, then $\exists x \in X$ such that $y \succ_{\text{tc}(v)} x$, this implies $z \succ_{\text{tc}(v)} x$ which is impossible from the above observation that $Z \cap Y = \emptyset$.

Consider all rankings $\tilde{\Omega}(v)$ where we place alternatives of Y at the top, X in the middle and Z at the bottom. Within Y and Z we rank alternatives respecting $\text{tc}(v)$ and since X is an anti-chain, rank these alternatives without restrictions. That is

$$\tilde{\Omega}(v) = \{r \mid \forall y \in Y, x \in X, z \in Z, y \succ_r x, x \succ_r z, r|_Y \in \Omega(\text{tc}(v)|_Y), r|_Z \in \Omega(\text{tc}(v)|_Z)\}.$$

Now we argue $\tilde{\Omega}(v) \subseteq \Omega(v)$. Note that we satisfy preference constraints when ranking within Y, X and Z . Also as we showed above, alternatives in Y are never dis-preferred to alternatives in X or Z and alternatives in X are never dis-preferred to alternatives in Z .

For the lower bound, first observe if $r \in \tilde{\Omega}(v)$ then $d(r, \sigma) = d(r|_Y, \sigma|_Y) + d(r|_X, \sigma|_X) + d(r|_Z, \sigma|_Z) + \delta$ where δ is defined in the theorem as the number of misorderings of alternatives across X, Y, Z , which is independent of r . Hence,

$$\sum_{r \in \Omega(v)} \phi^{d(r, \sigma)} \geq \sum_{r \in \tilde{\Omega}(v)} \phi^{d(r, \sigma)} = \phi^{\delta} \left[\sum_{r \in \Omega(\text{tc}(v)|_Y)} \phi^{d(r, \sigma|_Y)} \right] \left[\sum_{r \in \Omega(\text{tc}(v)|_X)} \phi^{d(r, \sigma|_X)} \right] \left[\sum_{r \in \Omega(\text{tc}(v)|_Z)} \phi^{d(r, \sigma|_Z)} \right].$$

Finally, it can be seen that the sum inside the third factor is exactly the normalization constant of an unconstrained Mallows model with $|X| = w(v)$ alternatives, and hence equal to $\prod_{i=1}^{w(v)} \sum_{j=0}^{i-1} \phi^j$, the second and fourth factors involve sums over rankings of Y and Z consistent with $\text{tc}(v)$. This proves the lower bound. \blacksquare

While the lower bound is not presented in a convenient closed-form, it is useful nonetheless if $w(v)$ is large: if there are few preference constraints in v (e.g., v involves only a small subset of alternatives) we expect $\Omega(v)$ to be large and hence have higher probability mass. We recover the true Mallows normalization constant if $v = \emptyset$ since $w(v) = m$. If v is highly constrained— $\Omega(v)$ has smaller probability mass—then $w(v)$ is small, but so are the factors involving summations in Equation 21. Note that ϕ^δ decreases as the number of comparisons in v that disagree with σ increases; this again corresponds to intuition.

With these bounds in hand, we can bound the quality of the posterior estimate $\hat{P}_v(r)$ produced by AMP:

Corollary 20 *Let L and U be the lower and upper bound as in Theorems 19 and 17, respectively. Then for $r \in \Omega(v)$, where l_i and h_i are defined in Proposition 11, we have*

$$\frac{L}{\prod_{i=1}^m \sum_{j=l_i}^{h_i} \phi^{i-j}} \leq \frac{\hat{P}_v(r)}{P_v(r)} \leq \frac{U}{\prod_{i=1}^m \sum_{j=l_i}^{h_i} \phi^{i-j}}. \tag{22}$$

Proof $\hat{P}_v(r)$ has the form given in Proposition 14 while $P_v(r) \propto \phi^{d(r,\sigma)}$. Then apply upper and lower bounds on the normalizing constant of $P_v(r)$. \blacksquare

3.2.4 MMP: AN MCMC SAMPLER BASED ON AMP

While AMP may have (theoretically) poor worst-case performance, we use it as the basis for a *statistically sound* sampler MMP, by exploiting AMP to propose new rankings for the Metropolis algorithm. With Equation 15, we can derive the acceptance ratio for Metropolis. At step $t + 1$ of Metropolis, let $r^{(t)}$ be the previous sampled ranking. Ranking r , proposed by AMP independently of $r^{(t)}$, will be accepted as the $t + 1$ st sample $r^{(t+1)}$ with probability $a^*(r, r^{(t)})$, where:

$$\begin{aligned} a^*(r, r^{(t)}) &= \min \left(1, \frac{\phi^{d(r,\sigma)} / Z_v}{\phi^{d(r^{(t)},\sigma)} / Z_v} \frac{\phi^{d(r^{(t)},\sigma)}}{\prod_{i=1}^m \phi^{i-h_i^t + \phi^{i-h_i^t+1} + \dots + \phi^{i-l_i^t}}} \right) \\ &= \min \left(1, \prod_{i=1}^m \begin{cases} \frac{h_i - l_i + 1}{h_i^t - l_i^t + 1} & \text{if } \phi = 1 \\ \frac{\phi^{h_i^t - h_i} (1 - \phi^{h_i - l_i + 1})}{1 - \phi^{h_i^t - l_i^t + 1}} & \text{otherwise} \end{cases} \right). \end{aligned} \tag{23}$$

Here the l_i s and h_i s are defined as in Equations 13 and 14, respectively (with respect to r ; and l_i^t), and h_i^t are defined similarly, but with respect to $r^{(t)}$. The term $Z_v = \sum_{r' \in \Omega(v)} \phi^{d(r',\sigma)}$ is the normalization constant of the Mallows posterior (given partial evidence v). The algorithm is specified in detail in Algorithm 2.

Exploiting Proposition 13, we can show:

Algorithm 2 MMP Sample Mallows Posterior using Metropolis

Input: v, σ, ϕ , number of steps T

- 1: **for** $t = 1..T$ **do**
- 2: $r \leftarrow \text{AMP}(v, \sigma, \phi)$
- 3: $a \sim \text{Uniform}[0,1]$
- 4: $r^{(t)} \leftarrow \begin{cases} r & \text{if } t = 1 \text{ or } a \leq a^*(r, r^{(t-1)}) \\ r^{(t-1)} & \text{otherwise} \end{cases}$
- 5: **end for**

Output: $r^{(T)}$

Theorem 21 *The Markov chain induced by MMP is ergodic on the class of states (rankings) $\Omega(v)$.*

Proof Note that the acceptance ratio as given in Equation 23 is always positive. The proposal distribution AMP draws rankings that are independent of previous rankings and by Proposition 13, its support is $\Omega(v)$. Hence, for any $r' \in \Omega(v)$, MMP has positive probability of making a transition to any ranking in $\Omega(v)$ —thus establishing that $\Omega(v)$ is a recurrent class—including itself—implying aperiodicity. ■

Thus, along with the detailed balance property of Metropolis, we have that the steady state distribution of MMP is exactly the Mallows posterior $P_v(r)$.

3.3 Sampling Mallows Mixture Posterior

Extending the GRIM, AMP and MMP algorithms to sampling from a mixture of Mallows models is straightforward. Recall the mixture posterior:

$$P(r, \mathbf{z} | v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \phi) = \frac{P(v|r, \alpha)P(r|\mathbf{z}, \boldsymbol{\sigma}, \phi)P(\mathbf{z}|\boldsymbol{\pi})}{\sum_{\mathbf{z}} \sum_{r \in \Omega} P(v|r, \alpha)P(r|\mathbf{z}, \boldsymbol{\sigma}, \phi)P(\mathbf{z}|\boldsymbol{\pi})}.$$

We use Gibbs sampling to alternate between r and \mathbf{z} , since the posterior does not factor in a way that permits us to draw samples exactly by sampling one variable, then conditionally sampling another. We initialize the process with some $\mathbf{z}^{(0)}$ and $r^{(0)}$, then repeatedly sample \mathbf{z} conditional on r , and r conditional on \mathbf{z} . For the t th sample, $\mathbf{z}^{(t)}$ is drawn from a multinomial with K outcomes:

$$\begin{aligned} P(\mathbf{z} : z_k = 1 | r^{(t-1)}, \boldsymbol{\pi}, \boldsymbol{\sigma}, \phi) &= \frac{P(r^{(t-1)} | \mathbf{z}, \boldsymbol{\sigma}, \phi)P(\mathbf{z} | \boldsymbol{\pi})}{\sum_{\mathbf{z}'} P(r^{(t-1)} | \mathbf{z}', \boldsymbol{\sigma}, \phi)P(\mathbf{z}' | \boldsymbol{\pi})} \\ &= \frac{\phi_k^{d(r^{(t-1)}, \sigma^{(k)})} \pi_k}{\sum_{k'=1}^K \phi_{k'}^{d(r^{(t-1)}, \sigma^{(k')})} \pi_{k'}}. \end{aligned}$$

To sample $r^{(t)}$ given \mathbf{z}^t , we use:

$$P(r | \mathbf{z}^{(t)}, v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \phi) = \frac{P(v|r)P(r|\mathbf{z}^{(t)}, \boldsymbol{\sigma}, \phi)P(\mathbf{z}^{(t)}|\boldsymbol{\pi})}{\sum_{r' \in \Omega} P(v|r')P(r'|\mathbf{z}^{(t)}, \boldsymbol{\sigma}, \phi)P(\mathbf{z}^{(t)}|\boldsymbol{\pi})}. \tag{24}$$

Algorithm 3 SP: Sample Mallows Mixture Posterior using Gibbs

Input: $v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}$, number of steps T

- 1: Initialize $r^{(0)}$ (e.g., topological sort on v)
- 2: **for** $t = 1..T$ **do**
- 3: $\mathbf{z}^{(t)} \sim P(\cdot | r^{(t-1)}, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}) \propto \phi_k^{d(r^{(t-1)}, \sigma^{(k)})} \pi_k$
- 4: Suppose $\mathbf{z}^{(t)}$ is indicator for k th component.
- 5: $r^{(t)} \leftarrow \text{AMP or MMP}(v, \sigma^{(k)}, \phi_k)$
- 6: **end for**

Output: $(\mathbf{z}^{(T)}, r^{(T)})$

Note that the term $P(\mathbf{z}^{(t)} | \boldsymbol{\pi})$ in the numerator and denominator cancels, and the missing completely at random assumption (see Equation 9) implies $P(v|r) = \mathbf{1}[r \in \Omega(v)]f(v)$, where f is a function independent of r . Thus Equation 24 becomes Equation 12 (conditioned on parameters $\sigma^{(k)}, \phi_k$). This is exactly the Mallows posterior sampling problem addressed in the previous section. Combining Gibbs sampling with sampling from a single component gives the overall SP algorithm, which is detailed in Algorithm 3. We note that this sampler is described using either MMP to exactly sample rankings (given the sampled mixture component) or AMP to allow more tractable, but approximate, sampling of rankings (see Line 5). In our experiments, we find that AMP works well within this Gibbs sampler.

4. EM Learning Algorithm for Mallows Mixtures

Armed with the sampling algorithms derived from GRIM, we now turn to maximum likelihood learning of the parameters $\boldsymbol{\pi}, \boldsymbol{\sigma}$, and $\boldsymbol{\phi}$ of a Mallows mixture using the *expectation maximization (EM)* algorithm. Before detailing our EM algorithm, we first consider the evaluation of the Mallows mixture log-likelihood in Section 4.1, which can be used to select the number of mixture components, or to test EM learning convergence. We then review the EM algorithm in Section 4.2 before detailing the steps of our EM learning procedure for Mallows mixture models in Section 4.3. In Section 4.4 we analyze the running time of our learning algorithm and suggest several ways to improve its performance.

4.1 Evaluating Log-Likelihood

The log-likelihood in our mixture model is

$$\mathcal{L}_\alpha(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} | V) = \sum_{\ell \in N} \ln \left[\sum_{\mathbf{z}_\ell} \sum_{r_\ell \in \Omega} P(v_\ell | r_\ell) P(r_\ell | \mathbf{z}_\ell, \boldsymbol{\sigma}, \boldsymbol{\phi}) P(\mathbf{z}_\ell | \boldsymbol{\pi}) \right]. \quad (25)$$

This can be rewritten as

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} | V) &= \sum_{\ell \in N} \ln \left[\sum_{k=1}^K \sum_{r_\ell \in \Omega(v_\ell)} \pi_k P(r_\ell | \sigma^{(k)}, \phi_k) \alpha^{|v_\ell|} (1 - \alpha)^{\binom{m}{2} - |v_\ell|} \right] \\ &= \sum_{\ell \in N} \ln \left[\sum_{k=1}^K \sum_{r_\ell \in \Omega(v_\ell)} \pi_k P(r_\ell | \sigma^{(k)}, \phi_k) \right] + \ln \left[\alpha^{|v_\ell|} (1 - \alpha)^{\binom{m}{2} - |v_\ell|} \right]. \end{aligned}$$

Note that the latter term involving α is decoupled from the other parameters, and in fact its maximum likelihood estimate is $\alpha^* = \sum_{\ell \in N} 2|v_\ell|/(nm(m-1))$. Since we are only interested in the log-likelihood as a function of the other parameters, we can ignore this additive constant and focus on

$$\mathcal{L}(\pi, \sigma, \phi | V) = \sum_{\ell \in N} \ln \left[\sum_{k=1}^K \sum_{r_\ell \in \Omega(v_\ell)} \frac{\pi_k \phi_k^{d(r_\ell, \sigma^{(k)})}}{Z_k} \right], \tag{26}$$

where Z_k is the Mallows normalization constant. Unfortunately, evaluating this term is provably hard.

Theorem 22 *Let $V = (v_1, \dots, v_n)$ be a profile of partial preferences. Computing the log-likelihood $\mathcal{L}(\pi, \sigma, \phi | V)$ is #P-hard.*

Proof We reduce the problem of counting the number of linear extensions of a partial order to this problem (see Theorem 9). Let v be a partial order for which we wish to count its linear extensions. We encode the input to log-likelihood computation as follows: let $V = (v)$, $K = 1$ with $\phi = 1$, and let σ be an arbitrary ranking. We have $\mathcal{L} = \mathcal{L}(\pi, \sigma, \phi | V) = \ln \sum_{r \in \Omega(v)} 1/m!$. Thus we can recover the number of linear extensions by computing $\exp(\mathcal{L}) \cdot m!$. That this can be accomplished in polynomial time can be seen by noting that \mathcal{L} is polynomial in m and we can use the power series expansion $\sum_{i \geq 0} \mathcal{L}^i m! / i!$, where we can truncating the series after a polynomial number of steps, after which the terms in the expansion no longer impact the integer portion of the solution (number of extensions). ■

Given the computational difficulty of evaluating the log-likelihood exactly, we consider approximations. We can rewrite the log-likelihood as

$$\mathcal{L}(\pi, \sigma, \phi | V) = \sum_{\ell \in N} \ln \left[\sum_{k=1}^K \pi_k \mathbb{E}_{P(r | \sigma^{(k)}, \phi_k)} \mathbf{1}[r \in \Omega(v_\ell)] \right],$$

and estimate the inner expectations by sampling from the Mallows model $P(r | \sigma^{(k)}, \phi_k)$. However, this can require exponential sample complexity in the worst case (e.g., if $K = 1$ and v is far from σ , i.e., $d(v, \sigma)$ is large, then to ensure v is in the sample requires a sample set of exponential size in expectation). But we can rewrite the summation inside the log as

$$\mathcal{L}(\pi, \sigma, \phi | V) = \sum_{\ell \in N} \ln \left[\sum_{k=1}^K \frac{\pi_k}{Z_k} \sum_{r \in \Omega(v_\ell)} \phi_k^{d(r, \sigma^{(k)})} \right],$$

and evaluate $\sum_{r \in \Omega(v_\ell)} \phi_k^{d(r, \sigma^{(k)})}$ using importance sampling:

$$\sum_{r \in \Omega(v_\ell)} \phi_k^{d(r, \sigma^{(k)})} = \mathbb{E}_{r \sim \hat{P}_{v_\ell}} \left[\frac{\phi_k^{d(r, \sigma^{(k)})}}{\hat{P}_{v_\ell}(r | \sigma^{(k)}, \phi_k)} \right]. \tag{27}$$

We generate samples $r_{\ell k}^{(1)}, \dots, r_{\ell k}^{(T)}$ with $\text{AMP}(v_\ell, \sigma^{(k)}, \phi_k)$ for $\ell \leq n$ and $k \leq K$, then substitute \hat{P}_v from Equation 15 into Equation 27 to obtain:

$$\sum_{\ell \in N} \ln \left[\sum_{k=1}^K \frac{\pi_k}{Z_k} \frac{1}{T} \sum_{t=1}^T \prod_{i=1}^m \sum_{j=i-h_i^{(\ell kt)}}^{i-l_i^{(\ell kt)}} \phi_k^j \right],$$

where $h_i^{(\ell kt)}$ and $l_i^{(\ell kt)}$ are defined in Equations 14 and 13, and defined with respect to $r_{\ell k}^{(t)}$, $\sigma^{(k)}$, and v_ℓ . We can simplify the expression inside the log and derive the estimate:

$$\hat{\mathcal{L}}(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} | V) = \sum_{\ell \in N} \ln \left[\frac{1}{T} \sum_{k=1}^K \sum_{t=1}^T \pi_k \cdot \begin{cases} \frac{1}{m!} \prod_{i=1}^m (h_i^{(\ell kt)} - l_i^{(\ell kt)} + 1) & \text{if } \phi_k = 1 \\ \phi_k^{\sum_{i=1}^m i - h_i^{(\ell kt)}} \prod_{i=1}^m \frac{1 - \phi_k^{h_i^{(\ell kt)} - l_i^{(\ell kt)} + 1}}{1 - \phi_k^i} & \text{otherwise} \end{cases} \right]. \quad (28)$$

As a matter of practical implementation, to ensure the sum of terms inside the log do not evaluate to zero (as it may be too small to be represented using common floating point standards), we observe that given numbers a and b with $a > b > 0$, $\ln(a + b) = \ln(a) + \ln(1 + b/a)$. Thus even if a and b are too small to be represented as floating point data types, we still obtain good approximations if $\ln(a)$ can be readily evaluated. This same technique can be used to ensure numerical stability.

4.2 The EM Algorithm

A popular approach to maximum likelihood estimation is the *expectation maximization* (EM) algorithm (Dempster et al., 1977). It is applied to probabilistic models in which a set of parameters θ determine the values of random variables, but observed data is available for only some of these variables. Let v denote the observed variables, and h the remaining unobserved (hidden or latent). In our model, we have $\theta = (\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}, \alpha)$, while v consists of a set of pairwise comparisons and $h = (\mathbf{z}, r)$ consist of the mixture-component assignment and its underlying complete preference ranking. EM is effectively a local search algorithm, which alternates between two steps. The *E-step* computes a posterior distribution over the hidden variables given the observed variables and a current estimate $\tilde{\theta}$ of the model parameters:

$$\mathbf{E}\text{-Step: } P(h|v, \tilde{\theta}).$$

The *M-step* computes, as its new estimate, those model parameters θ that maximize the expected value (w.r.t. $\tilde{\theta}$) of the log-likelihood (using the posterior computed in the E-step):

$$\mathbf{M}\text{-step: } \max_{\theta} \mathbb{E}_{P(h|v, \tilde{\theta})} \ln P(h, v | \theta).$$

These steps are iterated until convergence. Indeed, EM converges and gives a locally optimal solution, since each iteration of EM will increase the log-likelihood. In general one does not need to maximize the log-likelihood in the M-step, but simply increase it. An important

variation of EM called *Monte Carlo EM* is used when the posterior in the E-step is hard to compute (e.g., when dealing with large discrete event spaces, such as rankings). In Monte Carlo EM, one samples from the posterior in the E-step, and in the M-step simply optimizes the choice of parameters with respect to the empirical (sample) expectation.

4.3 Monte Carlo EM for Mallows Mixtures

Learning a Mallows mixture is challenging, since even evaluating its log-likelihood is #P-hard. A straightforward application of EM yields the following algorithm:

Initialization. Initialize values for $\boldsymbol{\pi}^{\text{old}}$, $\boldsymbol{\sigma}^{\text{old}}$, and $\boldsymbol{\phi}^{\text{old}}$.

E-step. Compute/estimate the posterior $P(\mathbf{z}_\ell, r_\ell | v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}})$ for all $\ell \in N$.

M-step. Compute model parameters that maximize expected log-likelihood:

$$\begin{aligned} \boldsymbol{\pi}^{\text{new}}, \boldsymbol{\sigma}^{\text{new}}, \boldsymbol{\phi}^{\text{new}} &= \operatorname{argmax}_{\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}} \sum_{\ell \in N} \mathbb{E}_{P(r_\ell, \mathbf{z}_\ell | v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}})} [\ln P(v_\ell, r_\ell, \mathbf{z}_\ell | \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi})] \\ &= \operatorname{argmax}_{\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}} \sum_{\ell \in N} \sum_{\mathbf{z}_\ell} \sum_{r_\ell \in \Omega} P(r_\ell, \mathbf{z}_\ell | v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}) \ln P(v_\ell, r_\ell, \mathbf{z}_\ell | \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}). \end{aligned}$$

Exact estimation in the E-step and optimization in the M-step is of course difficult due to the intractability of the Mallows posterior. Hence we resort to *Monte Carlo EM* and exploit our sampling methods to render EM tractable as follows. We initialize the parameters with values $\boldsymbol{\pi}^{\text{old}}$, $\boldsymbol{\sigma}^{\text{old}}$, and $\boldsymbol{\phi}^{\text{old}}$. For the E-step, instead of working directly with the posterior, we use GRIM-based Gibbs sampling (see Section 3.3) to obtain samples $(\mathbf{z}_\ell^{(t)}, r_\ell^{(t)})_{t=1}^T$ from the posteriors $P(r_\ell, \mathbf{z}_\ell | v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}})$ of each agent $\ell \leq n$. We note once again that Gibbs sampling may use either approximate AMP or the full-fledged MCMC MMP to generate rankings.

In the M-step, we maximize the expected log-likelihood using the *empirical expectation* with respect to the generated samples:

$$\boldsymbol{\pi}^{\text{new}}, \boldsymbol{\sigma}^{\text{new}}, \boldsymbol{\phi}^{\text{new}} = \operatorname{argmax}_{\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}} \sum_{\ell=1}^n \frac{1}{T} \sum_{t=1}^T \ln P(v_\ell, r_\ell^{(t)}, \mathbf{z}_\ell^{(t)} | \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}). \quad (29)$$

We show below in Theorem 23 that we can perform this maximization by adjusting the three (sets of) parameters in sequence—specifically, if the parameters are maximized in the order $\boldsymbol{\pi}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\phi}$ (and the first two can be maximized independently), this provides a globally optimal solution for the M-step (i.e., the solution obtained by optimizing parameters simultaneously). However, optimization of $\boldsymbol{\sigma}$, in particular, is NP-hard (as we discuss below), so we use a local search heuristic to approximate the choice of reference rankings in the M-step. We now detail the steps involved in the M-step optimization.

Somewhat abusing notation, let indicator vector $\mathbf{z}_\ell^{(t)}$ denote the mixture component to which the t th sample derived from preference ℓ belongs. We partition the collection of *all agent samples* (over all ℓ) into such classes: let $S_k = (\rho_{k1}, \dots, \rho_{kj_k})$ be the sub-sample of the rankings $r_\ell^{(t)}$, over all $\ell \in N, t \in [T]$, that are drawn from the k th component of the mixture model, i.e., where $\mathbf{z}_\ell^{(t)} = k$. Note that $j_1 + \dots + j_K = nT$. We can rewrite the

objective in the M-step as

$$\frac{1}{T} \sum_{k=1}^K \sum_{i=1}^{j_k} \ln P(v_{\ell(k,i)} | \rho_{ki}) P(\rho_{ki} | \sigma^{(k)}, \phi_k) P(k | \pi_k),$$

where $\ell(k, i)$ is the agent in sample $\rho_{k,i}$. We ignore $\ln P(v_{\ell(k,i)} | \rho_{ki})$, which only impacts α ; and we know $\rho_{ki} \in \Omega(v_{\ell(k,i)})$. Thus, we can rewrite the objective as

$$\sum_{k=1}^K \sum_{i=1}^{j_k} \left[\ln \pi_k + d(\rho_{ki}, \sigma^{(k)}) \ln \phi_k - \sum_{w=1}^m \ln \frac{1 - \phi_k^w}{1 - \phi_k} \right]. \tag{30}$$

where the last summation is the log of the Mallows normalization term.

OPTIMIZING π . We apply the method of Lagrange multipliers. The Lagrangian $L = (\sum_{k=1}^K \sum_{i=1}^{j_k} \ln \pi_k) + \lambda(\pi_1 + \dots + \pi_K - 1)$, where we have removed irrelevant terms of the objective not involving π . Taking the gradient, setting to zero and solving the system of equations $\nabla_{\pi, \lambda} L = 0$, we obtain:

$$\pi_k = \frac{j_k}{nT}, \forall k \leq K. \tag{31}$$

OPTIMIZING σ . The only term involving σ in Equation 30 is $\sum_{k=1}^K \sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)}) \ln \phi_k$. Since $\ln \phi_k$ is a negative scaling factor, and we can optimize the reference rankings $\sigma^{(k)}$ for each mixture component independently, we obtain:

$$\sigma^{(k)*} = \operatorname{argmin}_{\sigma^{(k)}} \sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)}). \tag{32}$$

Optimizing the choice of reference ranking $\sigma^{(k)}$ within a mixture component requires computation of the Kemeny ranking with respect to the rankings in S_k . This is, unfortunately, an NP-hard problem (Bartholdi III et al., 1989). To maintain tractability, we exploit the notion of *local Kemenization* (Dwork et al., 2001): instead of optimizing the ranking, we compute a locally optimal $\sigma^{(k)}$, in which swapping any two *adjacent* alternatives in $\sigma^{(k)}$ does not reduce the sum of distances in the Kemeny objective. While this may not result in optimal rankings, it has been shown to be extremely effective experimentally (Dwork et al., 2001; Busse et al., 2007).

We detail our local Kemenization algorithm in Algorithm 4. It works by first initializing the new ranking $\sigma^{(k)}$ to that from the previous EM iteration, $\sigma^{\text{old},(k)}$. Then, for each alternative x , starting with those at the top of the ranking and moving downwards, we evaluate swaps of x with the element above it, say y , and proceeding with the swap if the majority of rankings in S_k prefer x over y . This proceeds until the first potential swap of x fails (at which point we move on to the next alternative). This results in a locally optimal ranking (Dwork et al., 2001). Note we need not store all rankings in S_k ; we require only its *pairwise tournament graph*, which is a complete directed graph with vertices corresponding to the alternatives A and the weight of each edge $x \rightarrow y$ set to be $c_{xy} = |\{\rho \in S_k : y \succ_{\rho} x\}|$. Here c_{xy} is the “cost” of placing x above y .

Algorithm 4 LocalKemeny

Input: $S_k = (\rho_{k1}, \dots, \rho_{kj_k})$

- 1: $\sigma \leftarrow \sigma_k^{\text{old}}$
- 2: Compute pairwise tournament graph:
- 3: **for** all pair $(x, y) : x, y \in A$ and $x \neq y$ **do**
- 4: $c_{xy} = |\{\rho \in S_k : y \succ_{\rho} x\}|$.
- 5: **end for**
- 6: $d \leftarrow \sum_{\{x,y\} : x \succ_{\sigma^{(k)}} y} c_{xy}$
- 7: **for** $i = 2..m$ **do**
- 8: $x \leftarrow$ alternative in i th rank of σ
- 9: **for** $j = i - 1..1$ **do**
- 10: $y \leftarrow$ alternative in j th rank of σ
- 11: **if** $c_{xy} < c_{yx}$ **then**
- 12: Swap x with y
- 13: $d \leftarrow d - c_{xy} + c_{yx}$
- 14: **else**
- 15: quit this loop
- 16: **end if**
- 17: **end for**
- 18: **end for**

Output: σ , Kemeny cost d

OPTIMIZING ϕ . When optimizing ϕ in Equation 30, the objective decomposes into a sum that permits independent optimization of each ϕ_k . Exact optimization of ϕ_k is difficult; however, we can use gradient ascent with:

$$\frac{\partial (\text{Equation 30})}{\partial \phi_k} = \frac{d(S_k, \sigma^{(k)})}{\phi_k} - j_k \sum_{i=1}^m \frac{[(i-1)\phi_k - i]\phi_k^{i-1} + 1}{(1 - \phi_k^i)(1 - \phi_k)},$$

where $d(S_k, \sigma^{(k)}) = \sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)})$ is the Kemeny objective, which we obtain after running LocalKemeny.

Theorem 23 *Let π^* be given by Equation 31, σ^* be given by Equation 32, and ϕ^* be the optimal ϕ in Equation 30 where π is replaced with π^* and σ is replaced with σ^* . Then π^* , σ^* and ϕ^* is a globally optimal solution to Equation 29.*

Proof Regardless of the values of σ and ϕ , π is optimized by Equation 31 (see our analysis above), giving the optimal solution. It is also easy to see that the optimal reference rankings σ are the Kemeny rankings corresponding to ranking sets S_1, \dots, S_K , respectively, independent of the value of ϕ . Finally, if we substitute the optimal values π^* and σ^* into Equation 30, its optimal solution ϕ^* forms part of the optimal solution $(\pi^*, \sigma^*, \phi^*)$ to Equation 29. ■

It isn't difficult to see that a "locally optimal" pair (σ, ϕ) obtained by optimizing σ first, then ϕ is a locally optimal pair for Equation 29. Hence the resulting EM estimates

are also locally optimal with respect to the likelihood (Neal and Hinton, 1999). While no approximation bounds can be given, this lends some support to the optimization approach we adopt. To test the convergence of EM, one can test the convergence of the parameters (use Kendall-tau distance to measure σ against that of the previous iteration). One can also measure whether the log-likelihood is converging.

To reduce problems with local maxima, we initialize the mixture parameters using a K -means clustering approach where distances are measured using Kendall-tau rather than the usual squared Euclidean distance. One can use a modified version of Lloyd’s 1982 method for K -means, where the “centroid” (pertaining to Lloyd’s method) of a set of rankings can be simply its Kemeny ranking.

4.4 Complexity of EM Steps

We analyze the running time of one iteration of our EM approach. In the E-step, we sample variables (\mathbf{z}, r) . We need not store the ranking r for the component corresponding to \mathbf{z} , since in the M-step we do not need the actual rankings in S_k , but only its pairwise tournament graph. Hence we need only update the tournament graph corresponding to component \mathbf{z} with sample r , which takes $O(m^2)$ time. When sampling r , let T_{Metro} be the number of Metropolis steps before using the next sample. Each draw of r from AMP requires $O(m^2)$ time. Sampling \mathbf{z} requires $O(Km \log m)$ time since Kendall-tau distance can be computed in $O(m \log m)$ time. Let T_{Gibbs} be the number of Gibbs sampling steps run Gibbs before outputting a sample and suppose we restart Gibbs after each such sample. Suppose also we draw T_P posterior samples for each data point v_ℓ . Then the E-step takes $O(nT_P T_{\text{Gibbs}}(T_{\text{Metro}}m^2 + Km \log m))$ time. In practice, one can choose a very small number of samples, and run relatively few steps, when running the MCMC methods. Indeed, in our experiments below, we don’t use MMP within the Gibbs sampler, but instead use AMP directly (this can be viewed as running Metropolis for a single step); we discuss this further below. In principle, posterior sampling can be executed in parallel, with multiple processors handling the sampling and tournament graph updates for disjoint subsets of the data v_ℓ , with the results from different processors merged into the K tournament graphs.

For the M-step, updating π takes constant time, while updating the component reference rankings σ takes $O(Km^2)$ time. Optimizing ϕ can also be realized effectively, for instance, by using gradient ascent and bounding number of iterations. Hence the M-step requires $O(Km^2)$ time. Space complexity is dominated by the size of the K tournament graphs, hence is $O(Km^2)$.

Various techniques can be used to speed up computation from a practical perspective. Instead of storing the tournament graphs, which require quadratic memory, one can instead approximate the Kemeny ranking for any component using the *Borda count* to rank alternatives, which is a 5-approximation to Kemeny (Coppersmith et al., 2006), and often provides much better approximations in practice. If using Borda, when generating a complete ranking r in the posterior-sampling step (E-step) belonging to component k , one need only to update the Borda scores of all alternatives within component k ; in the M-step we simply rank alternatives (within each component) according to their sampled Borda scores. We still need the Kemeny distance between the resulting Borda ranking and the sampled rankings, but this can be approximated by re-running the E-step and evaluating the Kendall-tau

distance in an online fashion. One might also consider using Spearman footrule distance, which can be computed in $O(m)$ time rather than $O(m \log m)$ as in Kendall-tau, since it is 2-approximation to the Kemeny distance (Diaconis, 1988).

5. Experiments

We perform a series of experiments to validate the efficacy of our sampling and learning algorithms, to discover interesting properties of the learned mixture models on several popular data sets, and to evaluate the predictive power of our learned models to help predict missing preferences. We first assess the quality of our GRIM-based posterior sampling method AMP, measuring its accuracy relative to the true Mallows posterior. We then measure the approximation quality of our Monte Carlo algorithm for evaluating the Mallows mixture log-likelihood. Next we apply our EM algorithm to learn mixture models using several data sets: synthetically generated data sets, a Movielens ratings data set (with large m); and a sushi preference data set. The synthetic data experiments confirm the effectiveness of our EM algorithm while also revealing insights on how the size of preference data (either n or α) impacts learning. We also remark on some of its connections to crowdsourcing. Finally we assess the predictive accuracy of the learned models by conditioning on partial preference information and inferring the probability of the missing pairwise comparison preferences. In all experiments, we use Equation 26 to measure log-likelihood.

5.1 Sampling Quality

We first assess how well AMP approximates the true Mallows posterior P_v using randomly generated (synthetic) data. We vary parameters m , ϕ and α , while fixing a canonical reference ranking $\sigma = (1, 2, \dots, m)$. For each parameter setting, we generate 20 preferences v (e.g., the partial preferences of 20 agents) using our mixture model (see Section 2.3 and Equations 9 and 10), and evaluate the exact KL-divergence of P_v with respect to \hat{P}_v .⁵ This divergence is normalized by the entropy of P_v , since, when increasing m , KL-divergence and entropy both increase. Results are shown in Figure 3, with fixed and varying parameters for all three plots described in the caption. These results indicate that AMP approximates the posterior very well, with average normalized KL error ranging from 1–5%, across the parameter ranges tested.

5.2 Evaluating Log-Likelihood

In Section 4.1 we showed the #P-hardness of evaluating the log-likelihood and derived a Monte Carlo estimator that uses the AMP sampler. We evaluate the quality of the approximation produced by this estimator in this section. We vary three parameters to generate three experiments: (a) the number of alternatives m ; (b) the number of mixture components K ; and (c) the number of samples T per agent and per component (Equation 28). In all experiments, we fix the number of agents (i.e., the number of input preferences) at $n = 50$.

5. To compute KL-divergence, we need only consider consistent completions of our partial preferences. This set of rankings usually has size much smaller than $m!$, and can be enumerated by modifying the topological sort algorithm.

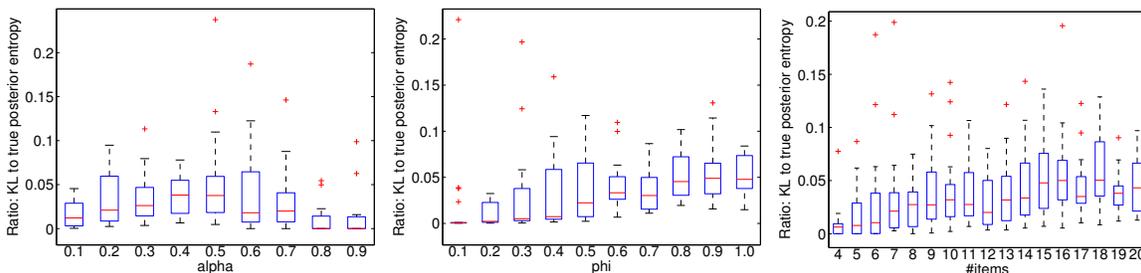


Figure 3: Comparing the posterior generated by AMP to the true Mallows posterior: normalized KL-divergence. The box-and-whisker plots have boxes shown the 25-75 percentile range over 20 runs, with the line inside box indicating the median, and the ‘+’ symbols outliers. From left to right: **Plot 1:** Varying α , while fixing $\phi = 0.5$, $m = 10$. **Plot 2:** Varying ϕ , while fixing $\alpha = 0.2$, $m = 10$. **Plot 3:** Varying m , fixing $\phi = 0.5$ and for $m \leq 13$, $\alpha = 0.2$, for $m > 13$, $\alpha = 0.5$.

In setting (a) (varying m), we generate v from a mixture model with $K = 3$ and $\pi = (1/3, 1/3, 1/3)$, $\phi = (1/2, 1/2, 1/2)$ and $\alpha = 0.2$. Each σ^k ($k \leq K$) is drawn uniformly at random from Ω .

In setting (b) (varying K), we generate v from a mixture model with K components, where $m = 8$, $\pi = (1/K, \dots, 1/K)$, $\phi = (1/2, \dots, 1/2)$ and $\alpha = 0.2$. Again σ drawn uniformly at random as in setting (a).

In setting (c) (varying T), parameters are $K = 1$, $m = 8$, σ chosen uniformly at random, $\phi = 0.5$ and $\alpha = 0.2$.

The parameters for which we evaluated the log-likelihood are generated as follows: mixture weights π are sampled from a “uniform” Dirichlet distribution with a parameter vector (i.e., equivalent sample size counts) consisting of K 5s. The reference rankings σ were drawn uniformly at random, and ϕ is drawn uniformly at random from interval $(0, 1)$.

The results for all three settings are shown in Figure 4. Overall we see that the Monte Carlo approximation is very good, and improves significantly while reducing variance as we increase the sample size for each agent’s log-likelihood (as captured by $K \cdot T$). Increasing m slightly degrades approximation quality, although it offers excellent estimates across the entire range of tested values.

5.3 EM Mixture Learning

We now evaluate our EM mixture-learning algorithms on the synthetic, Sushi and Movielens data sets.

5.3.1 SYNTHETIC DATA

Having empirically established that AMP provides good approximations to the true posterior, and that the log-likelihood can be closely approximated by importance sampling, we now evaluate how effective our EM algorithm is at recovering parameters in a controlled

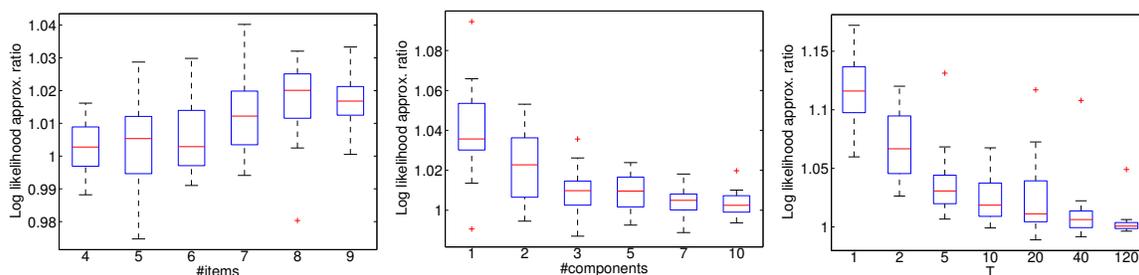


Figure 4: Comparing the ratio of the true log-likelihood to its Monte Carlo approximation. 20 instances are run per parameter setting. From left to right: **Plot 1:** Varying m , while fixing $T = 5$. **Plot 2:** Varying K while fixing $T = 5$. **Plot 3:** Varying T . Other parameter values are described in the text.

setting, using synthetic data generated from models with known parameters. We emphasize that the following experiments all use AMP within the Gibbs sampler in the E-step of Monte Carlo EM, rather than the MCMC algorithm MMP, given the approximation quality of AMP as well as its much better tractability.

We perform four experiments in which we vary: (a) α , the (expected) fraction of pairwise comparisons revealed from each preference; (b) the number of alternatives m ; (c) the number of mixture components K ; and (d) the number of agent preferences (data set size). In each experiment, we generate random model parameters as follows: π is drawn from a Dirichlet distribution with a uniform parameter vector of 5s; σ is drawn uniformly at random; and ϕ values are drawn uniformly at random from $[0.2, 0.8]$. Training data is generated using our probabilistic model with these parameters. When varying the single parameter for each experiment, we fix the other three, with fixed values: $\alpha = 0.2$, $m = 20$, $K = 3$ and $n = 50 \times K$. We analyze the performance of EM by (approximately) evaluating the ratio of the log-likelihood of the learned parameters to that of the true model parameters (π, σ, ϕ) on test data (preferences) generated from the true model—we set $n_{test} = n$ and $\alpha_{test} = 1$.

Results are shown in Figure 5 and provide some interesting insights. First they suggest that learning is more effective when either of α or n is larger (i.e., when we have more preference data for training). We also see that learning performance degrades when we increase the number of mixture components—this is hardly surprising, since there is less data per component as we increase K . Finally, learning improves as m increases for fixed values of α . This holds because the transitive closure for larger m tends to offer more preference information. For instance, $a_1 \succ a_2 \succ a_3 \succ a_4 \succ a_5 \succ a_6$ provides 5 comparisons, and corresponds to $1/9$ of all comparisons when $m = 10$, while leaving many comparisons unavailable, even after taking its transitive closure. By contrast, $a_1 \succ a_2 \succ \dots \succ a_{100}$ has 99 comparisons which is only $1/50$ of all comparisons when $m = 100$; but its transitive closure is a complete ranking.

These observations have interesting implications when considering information elicitation via “wisdom of the crowds.” When estimating a single objective ranking (i.e., $K = 1$), the amount of data needed for reliable estimation can be obtained by either increasing α (the

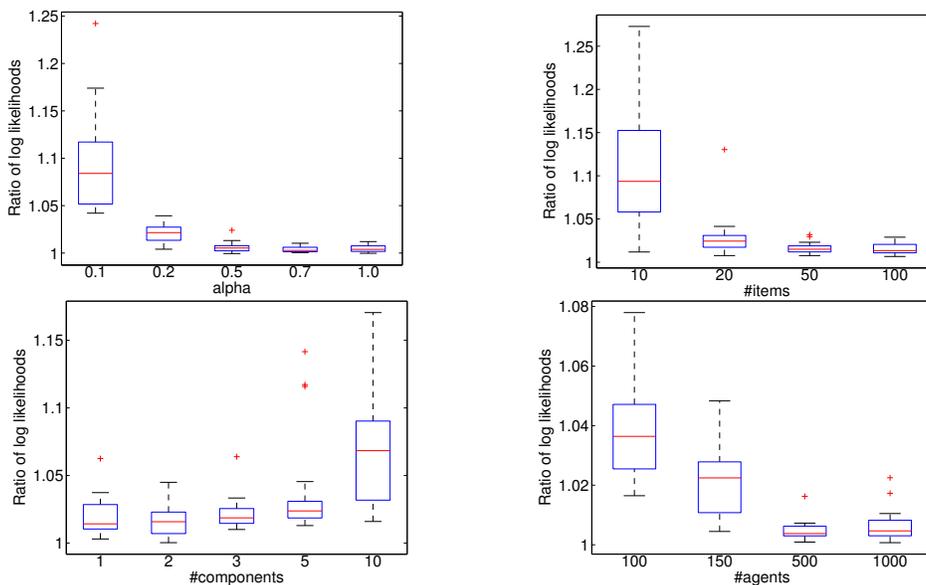


Figure 5: Performance of EM on synthetic data. Each plot shows the ratio of the log-likelihood of learned parameters to those of the true model parameters. Each parameter setting averages results of 20 instances. Log-likelihoods are approximated as in Section 4.1 with $T = 10$. Other parameter settings are described in the text.

average number of pairwise comparisons revealed per agent) and decreasing n (the number of agents queried) or by increasing n and decreasing α . In other words, one can obtain the same “effective” data by either asking more agents about their objective assessments while decreasing the number of questions per agent, as asking fewer agents to respond, but demanding more pairwise assessments per agent.

5.3.2 SUSHI DATA

The Sushi data set consists of 5000 complete rankings over 10 varieties of sushi indicating sushi preferences (Kamishima et al., 2005). We used 3500 preferences for training and 1500 for validation. We ran EM experiments by generating revealed pairwise comparisons for training with various probabilities α . To mitigate issues with local maxima, we ran EM ten times (more than is necessary) for each instance. Figure 6 shows that, even without complete preferences, EM learns well even with only 30-50% of all paired comparisons, though it degrades significantly at 20%, in part because only 10 alternatives are ranked (still performance at 20% is good when $K = 1, 2$). With $K = 6$ components, a good fit is found when training on complete preferences: Table 1 shows the learned clusters (all with reasonably low dispersion), illustrating interesting patterns (e.g., fatty tuna is strongly preferred by all but one group; a strong correlation exists across groups in preference/dispreference for salmon roe and sea urchin, which are “atypical fish”; and cucumber roll is consistently dispreferred).

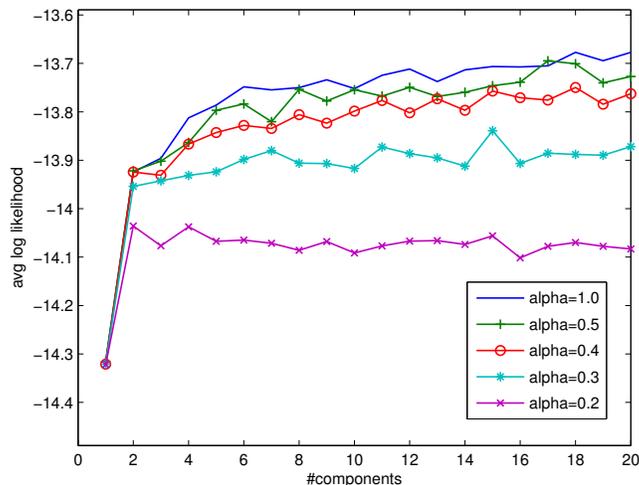


Figure 6: Sushi data set. Plots of average validation log-likelihood when the training data, pairwise comparisons, are revealed with probabilities $\alpha \in \{0.2, 0.3, 0.4, 0.5, 1.0\}$. Learning degrades as α gets closer to 0.2, that is, as more pairwise comparisons are removed.

| | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| $\pi_0 = 0.17$ | $\pi_1 = 0.15$ | $\pi_2 = 0.17$ | $\pi_3 = 0.18$ | $\pi_4 = 0.16$ | $\pi_5 = 0.18$ |
| $\phi_0 = 0.66$ | $\phi_1 = 0.74$ | $\phi_2 = 0.61$ | $\phi_3 = 0.64$ | $\phi_4 = 0.61$ | $\phi_5 = 0.62$ |
| fatty tuna | shrimp | sea urchin | fatty tuna | fatty tuna | fatty tuna |
| salmon roe | sea eel | fatty tuna | tuna | sea urchin | sea urchin |
| tuna | squid | sea eel | shrimp | tuna | salmon roe |
| sea eel | egg | salmon roe | tuna roll | salmon roe | shrimp |
| tuna roll | fatty tuna | shrimp | squid | sea eel | tuna |
| shrimp | tuna | tuna | sea eel | tuna roll | squid |
| egg | tuna roll | squid | egg | shrimp | tuna roll |
| squid | cucumber roll | tuna roll | cucumber roll | squid | sea eel |
| cucumber roll | salmon roe | egg | salmon roe | egg | egg |
| sea urchin | sea urchin | cucumber roll | sea urchin | cucumber roll | cucumber roll |

Table 1: Learned model for $K = 6$ on the sushi data set with complete preferences.

5.3.3 MOVIELENS DATA

We apply our EM algorithm to a subset of the Movielens data set (see www.grouplens.org) to find “preference types” across users. We use the 200 (out of roughly 3900) most frequently rated movies, and the ratings of the 5980 users (out of roughly 6000) who rated at least one of these. Integer ratings from 1 to 5 are converted to pairwise preferences in the obvious way (for ties, no preference was added to v). For example, if A and B had rating 5, C had rating 3 and D rating 1 then the user preference becomes $v = \{A \succ C, A \succ D, B \succ C, B \succ D, C \succ D\}$. We discard preferences that are empty when restricted to the top 200 movies, and use 3986 preferences for training and 1994 for validation. We run EM with number of components $K = 1, \dots, 20$; for each K we ran EM 20 times to mitigate the impact of local maxima. For each K , we evaluate average log-likelihood of the best run on the validation set to select the number of mixture components K . Log-likelihoods are approximated using our Monte Carlo estimator (with $K \cdot T = 120$).⁶

Log-likelihood results are shown in Figure 7 as a function of the number of mixture components. These results suggest that the best component sizes are $K = 10$ and $K = 5$ on the validation set. The learned model with $K = 5$ is displayed in Table 2, with each component ranking truncated to the top-20 movies. The five references rankings in this case are have some intuitive interpretation, but do not seem to exhibit the same separation as in the Sushi data set, in part due to the non-trivial overlap involving a number of “universally popular” movies (e.g., two movies, *The Shawshank Redemption* and *The Usual Suspects*, occur in all five components; two more occur in four, and more than 30 occur in three). Note also that the dispersion of each component is extremely high, approaching 1.

Despite this, certain patterns can be discerned. especially by focusing on reasonably *unique* movies, those than occur in only one or two components. For example, the second component contains the following “unique” movies: *Monty Python and the Holy Grail*, *The Maltese Falcon*, *Blade Runner*, *One Flew Over the Cuckoo’s Nest*, *A Clockwork Orange*, *2001: A Space Odyssey*, *North by Northwest*, *Pulp Fiction*, *Chinatown*, and *Apocalypse Now*. Themes within this cluster of unique movies include “older” science fiction, ultra-violence, actor Jack Nicholson and director Stanley Kubrick. The average date of the (top) twenty movies within this component is 1970, which is significantly lower than those of other components.

The same analysis of the fifth component shows the following “unique” movies: *A Christmas Story*, *This is Spinal Tap*, *American Beauty*, *Pulp Fiction*, *The Princess Bride*, *Forrest Gump*, *Fight Club*, *Fargo*, *Ferris Bueller’s Day Off*, *Raising Arizona*, *Good Will Hunting*, and *The Matrix*. Many of the movies here would commonly be characterized as “quirky,” including five “quirky comedies,” and several that tend toward extreme violence. The movies in this component also have a significantly later average date, 1992, than the others.

6. The C++ implementation of our algorithms have EM wall clock times of 15–20 minutes (Intel Xeon dual-core, 3GHz), certainly practical for a data set of this size. In other data sets, given the smaller number of alternatives, run times are much faster.

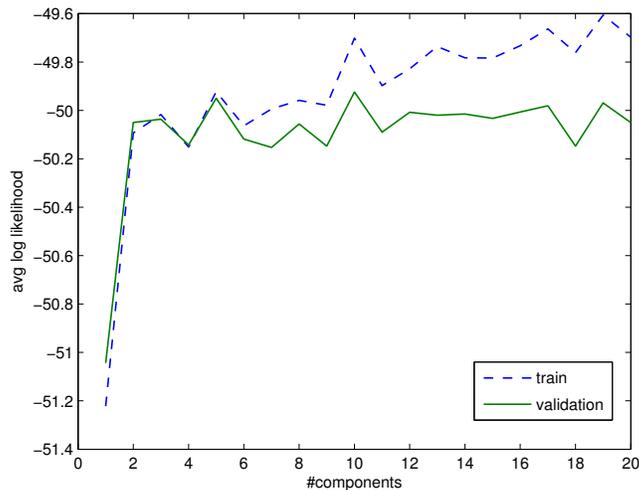


Figure 7: Movielens data set: average training and validation log likelihoods on the learned model parameters of different component sizes.

| $\pi_1 = 0.24, \phi_1 = 0.98$ | $\pi_2 = 0.23, \phi_2 = 0.98$ | $\pi_3 = 0.21, \phi_3 = 0.98$ | $\pi_4 = 0.19, \phi_4 = 0.98$ | $\pi_5 = 0.13, \phi_5 = 0.97$ |
|--|---|---|---|----------------------------------|
| Citizen Kane (1941) | Godfather, The (1972) | Raiders of the Lost Ark (1981) | Shawshank Redemption, The (1994) | Usual Suspects, The (1995) |
| Godfather, The (1972) | Dr. Strangelove (1963) | Godfather, The (1972) | Life Is Beautiful (1997) | Shawshank Redemption, The (1994) |
| Dr. Strangelove (1963) | Citizen Kane (1941) | Schindler's List (1993) | Raiders of the Lost Ark (1981) | Schindler's List (1993) |
| Schindler's List (1993) | Casablanca (1942) | Rear Window (1954) | Schindler's List (1993) | Life Is Beautiful (1997) |
| Rear Window (1954) | Star Wars: Episode IV - A New Hope (1977) | Star Wars: Episode IV - A New Hope (1977) | Star Wars: Episode IV - A New Hope (1977) | Christmas Story, A (1983) |
| Shawshank Redemption, The (1994) | Usual Suspects, The (1995) | Shawshank Redemption, The (1994) | Matrix, The (1999) | This Is Spinal Tap (1984) |
| American Beauty (1999) | Raiders of the Lost Ark (1981) | Casablanca (1942) | Sixth Sense, The (1999) | American Beauty (1999) |
| Godfather: Part II, The (1974) | Monty Python and the Holy Grail (1974) | Sixth Sense, The (1999) | Sting, The (1973) | Sixth Sense, The (1999) |
| One Flew Over the Cuckoo's Nest (1975) | Rear Window (1954) | Psycho (1960) | Forrest Gump (1994) | Pulp Fiction (1994) |
| Casablanca (1942) | Maltese Falcon, The (1941) | Citizen Kane (1941) | Usual Suspects, The (1995) | Princess Bride, The (1987) |
| Usual Suspects, The (1995) | Blade Runner (1982) | Sting, The (1973) | Braveheart (1995) | Silence of the Lambs, The (1991) |
| Pulp Fiction (1994) | One Flew Over the Cuckoo's Nest (1975) | Usual Suspects, The (1995) | Green Mile, The (1999) | Godfather, The (1972) |
| Monty Python and the Holy Grail (1974) | Clockwork Orange, A (1971) | Saving Private Ryan (1998) | Indiana Jones and the Last Crusade (1989) | Forrest Gump (1994) |
| Fargo (1996) | 2001: A Space Odyssey (1968) | Godfather: Part II, The (1974) | Saving Private Ryan (1998) | Fight Club (1999) |
| Life Is Beautiful (1997) | North by Northwest (1959) | Silence of the Lambs, The (1991) | Princess Bride, The (1987) | Fargo (1996) |
| Graduate, The (1967) | Pulp Fiction (1994) | Wizard of Oz, The (1939) | Star Wars: Episode V - The Empire Strikes Back (1980) | Ferris Bueller's Day Off (1986) |
| North by Northwest (1959) | Godfather: Part II, The (1974) | Dr. Strangelove (1963) | Silence of the Lambs, The (1991) | Raising Arizona (1987) |
| GoodFellas (1990) | Chinatown (1974) | Jaws (1975) | Good Will Hunting (1997) | Saving Private Ryan (1998) |
| Chinatown (1974) | Apocalypse Now (1979) | Braveheart (1995) | Ferris Bueller's Day Off (1986) | Good Will Hunting (1997) |
| Raiders of the Lost Ark (1981) | Shawshank Redemption, The (1994) | Aliens (1986) | When Harry Met Sally (1989) | Matrix, The (1999) |

Table 2: Learned model for $K = 5$ on Movielens. Shows the top 20 (out of 200) movies.

5.4 Predicting Missing Pairwise Preferences

In our prediction experiments, we seek to evaluate the performance of the learned models in predicting unseen pairwise comparisons. In particular, we use the complete sushi data set, train our mixture model on the first 3500 complete rankings (we train for all $K = 1, \dots, 20$), and select the best K by evaluating the log-likelihood on the validation data set, which consists of 500 complete rankings. It turns out that a mixture model with $K = 6$ was most suitable.

To test posterior prediction performance, we use 1000 complete rankings, distinct from both the training and validation sets, and randomly remove a fraction $1 - \alpha$ of the pairwise comparisons from each ranking, then compute the transitive closure of the remaining comparisons to obtain partial preferences. We generate preferences for four different values of α . With $\alpha = 0$, all preferences are removed; with $\alpha = 0.25$, 42% of the pairwise comparisons are left after computing transitive closures; with $\alpha = 0.5$, 76% of the all pairwise comparisons remain; and with $\alpha = 0.75$, 83% of the pairwise comparisons are left.

We conditioned the learned model on the partial preferences of each agent in turn, to obtain posterior distributions over which we can infer each agent’s missing pairwise comparisons. In making predictions, we use our posterior sampling algorithm SP to sample complete rankings, which we then use to update a tournament graph—recall, this is a set of counts c_{ab} to count the number of rankings for which $a \succ b$, for all $a, b \in A$. Then we estimate the posterior probability $P(a \succ b \mid v)$ by $\frac{c_{ab}}{c_{ab} + c_{ba}}$.

We define our prediction loss as follows. Suppose we have a complete ranking r with its corresponding partial preference v obtained as described above. For a given $a \succ_r b$ that is unobserved in $\text{tc}(v)$, we define the posterior prediction loss to be $\hat{P}(a \prec b \mid v) = \frac{c_{ba}}{c_{ab} + c_{ba}}$. Let $M(v) = \{(a, b) : a \succ_r b, a \succ b \notin \text{tc}(v)\}$ be the set of missing pairwise comparisons in v . We define the *average loss of v* as

$$\varepsilon_v = \frac{\sum_{(a,b) \in M(v_\ell)} \hat{P}(a \prec b \mid v)}{|M(v_\ell)|}.$$

We next define the *average loss per preference* to be

$$\bar{\varepsilon} = \frac{1}{n} \sum_{\ell=1}^n \varepsilon_{v_\ell},$$

where n is the number of distinct agents or preferences (in this case $n = 1000$). For $a \succ_r b$, let $\mathcal{D}(a, b) = r(b) - r(a)$ be the difference in their rank positions and $M_D(v) = \{(a, b) \in M(v) : \mathcal{D}(a, b) = D\}$. We also measure the *average loss at distance D* as follows:

$$\bar{\varepsilon}_D = \frac{\sum_{\ell=1}^n \sum_{(a,b) \in M_D(v_\ell)} \hat{P}(a \prec b \mid v)}{\sum_{\ell=1}^n |M_D(v_\ell)|}.$$

The results for average loss per preference are as follows:

- $\bar{\varepsilon} = 0.43$ for preferences generated with $\alpha = 0$;
- $\bar{\varepsilon} = 0.35$ for $\alpha = 0.25$,
- $\bar{\varepsilon} = 0.39$ for $\alpha = 0.5$, and

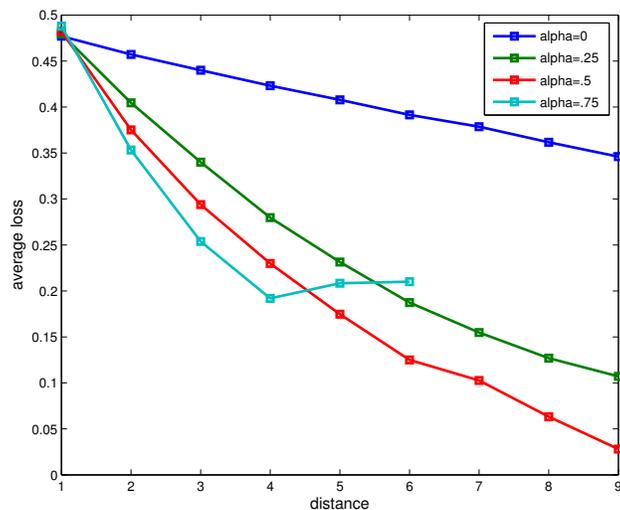


Figure 8: Sushi prediction results: average prediction loss for missing pairwise comparisons for pairs at different distances in the underlying ranking.

- $\bar{\epsilon} = 0.44$ for $\alpha = 0.75$.

(We interpret these results below). Results for $\bar{\epsilon}_D$ at various distances are plotted in Figure 8. Since these results are extremely sensitive to the number of pairwise comparisons available in the data at different distances, we show the number of such comparisons, per distance, in Table 3.

The results indicate that predictive performance is weakly accurate when pairs are close in distance, but improves as the distance between the predicted alternatives increases in the underlying ranking. For $\alpha = 0.75$, the average loss at distances 5 and 6 is higher than expected, but this is due to the small number of comparisons missing available for testing (and in general) at those distances. We also observe that the number of comparisons of a particular distance decreases as a function of the distance—this is more pronounced for smaller values of α . This can be attributed to the use of transitive closure: the further apart a pair of alternatives are in the underlying ranking, the less likely it is that we will remove all of the pairwise comparisons required render the two alternatives incomparable *after* taking the transitive closure of the preferences that remain. As a consequence of the skewed distribution of missing pairs available for prediction at specific distances, the average loss per preference does not in fact decrease as α increases. For example, it is 0.39 for $\alpha = 0.5$, and 0.44 for $\alpha = 0.75$; this is because the *relative* number of missing comparisons at smaller distances (which are more difficult to predict) is much greater when $\alpha = 0.75$ than when $\alpha = 0.5$ (as shown in Table 3).

6. Applications to Non-Parametric Estimators

Lebanon and Mao (2008) propose a non-parametric estimator for Mallows models when observations form partitioned preferences. This estimator is an analogue of typical kernel

| | $D = 1$ | $D = 2$ | $D = 3$ | $D = 4$ | $D = 5$ | $D = 6$ | $D = 7$ | $D = 8$ | $D = 9$ |
|----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| $\alpha = 0$ | 9000 | 8000 | 7000 | 6000 | 5000 | 4000 | 3000 | 2000 | 1000 |
| $\alpha = .25$ | 6610 | 5487 | 4393 | 3459 | 2492 | 1744 | 1092 | 606 | 256 |
| $\alpha = .5$ | 4429 | 2911 | 1683 | 898 | 443 | 191 | 65 | 22 | 1 |
| $\alpha = .75$ | 2230 | 824 | 244 | 62 | 9 | 2 | 0 | 0 | 0 |

Table 3: The number of missing pairwise comparisons (over all agents) among pairs that are distance D from each other, with preferences generated by randomly deleting fraction $1 - \alpha$ of preferences, then taking the transitive closure of the remaining comparisons.

density estimators, but over the space of rankings. Their purpose, similar to mixture models, is to model the distribution of real ranking data. The idea is to place “smooth unimodal bumps,” formulated as a single Mallows model, at every input (training) preference. This is much like a mixture model with the number of components equal to the number of preferences in the training data. They offer closed-form solutions by exploiting the existence of the closed-form for the Mallows normalization constant when partitioned preferences are observed. Unfortunately, with general pairwise comparisons, computing this normalization constant is intractable unless $\#P=P$. In contrast to our contributions above, they do not address the question of how to find a maximum likelihood estimate of the Mallows dispersion parameter, also known as the kernel width, which they suggest as being “extremely difficult.”

It turns out we can use AMP for approximate marginalization to support non-parametric estimation with general preference data. This shows the potential applicability of our sampling algorithm to a wider range of problems where observations consist of pairwise comparisons. We illustrate its application by defining a non-parametric estimator and deriving a Monte Carlo evaluation formula suitable for incomplete preferences.

Define a joint distribution q_ℓ over the probability space $\Omega(v_\ell) \times \Omega$:

$$q_\ell(s, r) = \frac{\phi^{d(r,s)}}{|\Omega(v_\ell)|Z_\phi}, \tag{33}$$

where Z_ϕ is the Mallows normalization constant with respect to dispersion ϕ . This distribution corresponds to drawing a ranking s uniformly at random from $\Omega(v_\ell)$ and then drawing r from a Mallows distribution with reference ranking s and dispersion ϕ . The estimator, extended in the style of Lebanon and Mao (2008) to any set of paired comparisons, is simply:

$$\begin{aligned} p(v) &= \frac{1}{n} \sum_{\ell \in N} q_\ell(s \in \Omega(v_\ell), r \in \Omega(v)) \\ &= \frac{1}{n} \sum_{\ell \in N} \sum_{s \in \Omega(v_\ell)} \sum_{r \in \Omega(v)} \frac{\phi^{d(r,s)}}{|\Omega(v_\ell)|Z_\phi}. \end{aligned} \tag{34}$$

Note that this is a distribution over *rankings* and not incomplete preferences, that is, a marginal over $\Omega(v)$. A special case arises when V consists entirely of complete rankings,

which simplifies to a mixture of Mallows models with n equally weighted components, each with one of the observed rankings v_ℓ as its reference ranking, and dispersion ϕ . This estimator can be useful for inference over the posterior $p(r|v) = p(r)\mathbf{1}[r \in \Omega(v)]/p(v)$ for $r \in \Omega(v)$. For any fixed v , let $f(s) = \sum_{r \in \Omega(v)} \phi^{d(r,s)}$. Then we have

$$\begin{aligned} p(v) &= \frac{1}{nZ_\phi} \sum_{\ell \in N} \sum_{s \in \Omega(v_\ell)} \frac{1}{|\Omega(v_\ell)|} f(s) \\ &= \frac{1}{nZ_\phi} \sum_{\ell \in N} \mathbb{E}_{s \sim \Omega(v_\ell)} f(s), \end{aligned}$$

where s is drawn uniformly from $\Omega(v_\ell)$. One can estimate the expectation by importance sampling. Suppose we draw, for each ℓ , rankings $s_\ell^{(1)}, \dots, s_\ell^{(T)}$ using AMP($v_\ell, \sigma, \phi = 1$) to approximate uniform sampling (e.g., choose some ranking σ from $\Omega(v_\ell)$). Let $w_{\ell t} = 1/\hat{P}_{v_\ell}(s_\ell^{(t)})$, which has a closed-form given by Equation 15. Then the estimate is

$$\hat{p}(v) = \frac{1}{nZ_\phi} \sum_{\ell=1}^n \frac{\sum_{t=1}^T w_{\ell t} f(s_\ell^{(t)})}{\sum_{t=1}^T w_{\ell t}}.$$

Evaluating $f(s_\ell^{(t)})$ is generally intractable, but again, it can be approximated using our earlier techniques, as given by Equation 27. In summary, we can realize non-parametric estimation using a nested sampling procedure to first approximate the outer expectation over s , followed by the inner summation $f(s)$.

7. Conclusion and Future Work

We have developed a set of algorithms to support the efficient and effective learning of ranking or preference distributions when the observed data comprise a set of unrestricted pairwise comparisons of alternatives. Given the fundamental nature of pairwise comparisons in revealed preference, our methods extend the reach of rank learning in a vital way. One of our main technical contributions, the GRIM algorithm, allows sampling of arbitrary distributions, including Mallows models conditioned on pairwise data. It supports a tractable approximation to the #P-hard problem of log-likelihood evaluation of Mallows mixtures; and it forms the heart of an EM algorithm that was shown to be quite effective in our experiments. GRIM can also be used for non-parametric estimation.

We are pursuing a number of interesting directions, including various extensions and applications of the model we have developed here. One of the weaknesses with Mallows is its lack of flexibility in various dimensions, such as allowing different dispersion “rates” in different regions of the ranking. Models that allow more flexibility while controlling for overfitting could lead to more realistic ranking models for real-world settings. Other extensions include exploration of other probabilistic models of incomplete preferences that employ different distributions over rankings, such as Plackett-Luce or weighted Mallows; that account for noisy comparison data from users; and that account for data that is not missing at random—this may occur, say, in settings in which a bias exists towards observing preferences for higher ranked alternatives.

In another vein, we are interested in exploiting learned preference models of the type developed here for decision-theoretic tasks in social choice or personalized recommendation. Learned preferences can be leveraged in both active *preference elicitation* (e.g., in social choice or group decision making (Lu and Boutilier, 2011)), or in passive (purely observational) settings. It would also be interesting to apply GRIM to other posterior distributions such as energy models, and to compare it to different MCMC techniques like chain flipping (Dellaert et al., 2003).

Acknowledgments

This research was supported by the the Natural Sciences and Engineering Research Council (NSERC) of Canada. Tyler Lu was supported by a Google Ph.D. Fellowship.

References

- Shivani Agarwal and Dan Roth. Learnability of bipartite ranking functions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory (COLT-05)*, pages 16–31, 2005.
- John Bartholdi III, Craig Tovey, and Michael Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.
- Christophe Biernacki and Julien Jacques. A generative model for rank data based on insertion sort algorithm. *Computational Statistics and Data Analysis*, 58:162–176, 2013.
- Graham Brightwell and Peter Winkler. Counting linear extensions is $\#P$ -complete. In *ACM Symposium on the Theory of Computing*, pages 175–181, 1991.
- Chris J.C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, July 2010.
- Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the Twenty-Second International Conference on Machine Learning (ICML-05)*, pages 89–96, Bonn, Germany, 2005.
- Róbert Busa-Fekete, Eyke Hüllermeier, and Balázs Szörényi. Preference-based rank elicitation using statistical models: The case of Mallows. In *Proceedings of the Thirty-first International Conference on Machine Learning (ICML-14)*, pages 1071–1079, Beijing, China, 2014.
- Ludwig M. Busse, Peter Orbanz, and Joachim M. Buhmann. Cluster analysis of heterogeneous rank data. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning (ICML-07)*, pages 113–120, Corvallis, OR, 2007.
- Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Label ranking methods based on the Plackett-Luce model. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML-10)*, pages 215–222, Haifa, Israel, 2010.

- William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.
- marquis de (Marie Jean Antoine Nicolas de Caritat) Condorcet. *Essai sur l'Application de l'Analyse a la Probabilite des Decisions rendues a la Probabilite des Voix*. Paris: L'Imprimerie Royale, 1785.
- Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 145–152, Edinburgh, 2005.
- Vincent Conitzer, Matthew Rognlie, and Lirong Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 109–115, Pasadena, CA, 2009.
- Don Coppersmith, Lisa Fleischer, and Atri Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-06)*, pages 776–782, Miami, FL, 2006.
- Frank Dellaert, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50(1–2):45–71, 2003.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- Persi Diaconis. *Group Representations in Probability and Statistics*, volume 11 of *Lecture Notes–Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1988.
- Jean-Paul Doignon, Aleksandar Pekec, and Michel Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69(1):33–54, 2004.
- Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International World Wide Web Conference (WWW-01)*, pages 613–622, Hong Kong, 2001. ACM.
- M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):359–369, 1986. ISSN 00359246.
- Michael A. Fligner and Joseph S. Verducci, editors. *Probability Models and Statistical Analysis for Ranking Data*, volume 80 of *Lecture Notes in Statistics*. New York: Springer-Verlag, 1993.
- Brian Francis, Regina Dittrich, Reinhold Hatzinger, and Les Humphreys. A mixture model for longitudinal partially ranked data. *Communications in Statistics — Theory and Methods*, 43(4):722–734, 2014.

- Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.
- Zoubin Ghahramani and Mike I. Jordan. Learning from incomplete data. Technical report, Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab, 1995.
- Isobel Claire Gormley and Thomas Brendan Murphy. A latent space model for rank data. In Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, Anna Goldenberg, Eric P. Xing, and Alice X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, pages 90–102. Springer, 2007.
- Isobel Claire Gormley and Thomas Brendan Murphy. Exploring voting blocs within the Irish electorate. *Journal of the American Statistical Association*, 103(483):1014–1027, 2008.
- John Guiver and Edward Snelson. Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the Twenty-Sixth International Conference on Machine Learning (ICML-09)*, pages 377–384, Montreal, 2009. ISBN 978-1-60558-516-1.
- Jonathan Huang and Carlos Guestrin. Riffled independence for ranked data. In *Advances in Neural Information Processing Systems 21*, pages 799–807, Vancouver, 2009.
- Jonathan Huang, Ashish Kapoor, and Carlos Guestrin. Riffled independence for efficient inference with partial ranking. *Journal of Artificial Intelligence Research*, 44:491–532, 2012.
- Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16–17):1897–1916, 2008.
- Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.
- Toshihiro Kamishima, Hideto Kazawa, and Shotaro Akaho. Supervised ordering: An empirical survey. In *IEEE International Conference on Data Mining (ICDM-05)*, pages 673–676, Houston, TX, 2005.
- John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.
- Guy Lebanon and Yi Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.
- Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.
- Jordan J. Louviere, David A. Hensher, and Joffre D. Swait. *Stated Choice Methods: Analysis and Application*. Cambridge University Press, Cambridge, 2000.
- Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 287–293, Barcelona, 2011.

- R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.
- Colin L. Mallows. Non-null ranking models. In *Biometrika*, volume 44, pages 114–130, 1957.
- John I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, London, 1995.
- Benjamin M. Marlin and Richard S. Zemel. Collaborative filtering and the missing at random assumption. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 50–54, Vancouver, 2007.
- Marina Meila and Harr Chen. Dirichlet process mixtures of generalized Mallows models. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 358–367, Catalina Island, CA, USA, 2010.
- Thomas Brendan Murphy and Donal Martin. Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis*, 41:645–655, January 2003.
- Radford Neal and Geoff Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, Cambridge, MA, 1999.
- R. Plackett. The analysis of permutations. *Applied Statistics*, 24:193–202, 1975.
- T. Graepel R. Herbrich and K. Obermayer. Large margin rank boundaries for ordinal regression. In Peter J. Bartlett, Bernhard Schölkopf, Dale Schuurmans, and Alex J. Smola, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.
- Cynthia Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, 2009.
- Grace S. Shieh. A weighted Kendall’s tau statistic. *Statistics and Probability Letters*, 39 (1):17–24, 1998.
- Maksims Volkovs and Richard S. Zemel. Boltzrank: Learning to maximize expected ranking gain. In *Proceedings of the Twenty-Sixth International Conference on Machine Learning (ICML-09)*, page 137, Montreal, 2009.
- Maksims Volkovs and Richard S. Zemel. Efficient sampling for bipartite matching problems. In *Advances in Neural Information Processing Systems 25 (NIPS-12)*, pages 1322–1330, Lake Tahoe, NV, 2012.
- Peyton Young. Optimal voting rules. *Journal of Economic Perspectives*, 9:51–64, 1995.