

Workload-Driven Learning of Mallows Mixtures with Pairwise Preference Data

Julia Stoyanovich*
Drexel University
stoyanovich@drexel.edu

Lovro Ilijasic
Drexel University
lovro@drexel.edu

Haoyue Ping
Drexel University
hp354@drexel.edu

ABSTRACT

In this paper we present a framework for learning mixtures of Mallows models from large samples of incomplete preferences. The problem we address is of significant practical importance in social choice, recommender systems, and other domains where it is required to aggregate, or otherwise analyze, preferences of a heterogeneous user base.

We improve on state-of-the-art methods for learning mixtures of Mallows models with pairwise preference data. Exact sampling from the Mallows posterior in presence of arbitrary pairwise evidence is known to be intractable even for a single Mallows. This motivated to the development of an approximate sampler called AMP. In this paper we propose AMPx, an ensemble method for approximate sampling from the Mallows posterior that combines AMP with frequency-based estimation of posterior probabilities. We experimentally demonstrate that AMPx achieves faster convergence and higher accuracy than AMP alone. We also adapt state-of-the-art clustering techniques that have not been used in this setting, for learning parameters of the Mallows mixture, and show experimentally that mixture parameters can be learned accurately and efficiently.

1. INTRODUCTION

Preference data consists of orders among two or more items and arises in a broad range of domains, including social and scientific. Preferences come in a variety of forms, such as total orders or rankings, partial orders, top- k lists, and pair-wise comparisons. Over the past decade, there has been a sharp increase in the volume of preference data, in the diversity of applications that use it, and in the richness of preference data analysis methods. For example, in preferential voting systems used in elections, there is often a need to *correlate* socio-demographic information of voters with their ranking behavior [9], analysis that is useful to political campaigns. E-commerce and social applications may need

*This work was supported in part by NSF Grants No. 1464327 and 1539856, and BSF Grant No. 2014391.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WebDB'16, June 26-July 01 2016, San Francisco, CA, USA

© 2016 ACM. ISBN 978-1-4503-4310-7/16/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2932194.2932202>

to *cluster* users based on similarity in their preferences, and to use such clusters to describe properties of the user base and to produce recommendations.

The focus of this paper is on concisely describing preferences of a diverse population of users based on a *noisy incomplete sample*. We follow the realistic assumption that is made in much recent literature [2, 4, 8, 10, 15] that users whose preferences are represented in the sample are of different types. Our goal is thus to learn a population model that explains the diversity in the sample. More specifically, and also in-line with much recent literature [2, 15], we assume that the sample at hand is generated by a *mixture of Mallows models*, and that each user's preferences are drawn from one of the Mallows. We aim to estimate parameters of the mixture, that is, to discover the number of Mallows and their relative proportions, and to characterize each Mallows in the mixture by estimating its parameters.

Contributions. In this paper we improve on state-of-the-art methods for learning Mallows models with pairwise preference data, proposed in [15]. (1) We propose AMPx, an ensemble-based methods for approximate sampling from the Mallows posterior. We show that AMPx gives more accurate estimates than AMP alone, a method from [15] that AMPx uses as a subroutine, and that this leads to faster convergence in an EM-based framework. (2) We explore different versions of AMPx, and show that best accuracy is achieved when frequency information is updated iteratively, from round to round of EM, and further refined dynamically within a round. (3) We show how affinity propagation, a scalable clustering technique that has not been applied in this setting before, can be used to learn parameters of the Mallows mixture. (4) We conduct an experimental evaluation with real and synthetic datasets, showing that our approach is accurate and efficient.

Methods developed here form part of DB4Pref [12], an ambitious framework for end-to-end management, querying and analysis of preference data. Results of this work, including AMPx code, datasets, a powerful and flexible generator of synthetic data, repeatable experimental methodology, and experimental results, are available at DB4Pref.com.

Roadmap. In Section 2 we introduce the required notation, give an overview of the Mallows model, and explain why Mallows inference in presence of evidence is challenging. We describe our approach in Section 3, and validate it experimentally using real and synthetic datasets in Section 4. We present related work in Section 5 and conclude in Section 6.

2. PRELIMINARIES

Basic notation. We follow the notation used in [15] and elsewhere, and assume that n voters $V = v_1, \dots, v_n$ are stating their preferences with respect to m alternatives $A = a_1, \dots, a_m$. We further assume that if a voter’s preferences were fully revealed, they would correspond to a ranking, or a total order, over the items in A . A ranking $\sigma : A \rightarrow [m]$ is a bijection, where $\text{rank}_\sigma(a)$ is the *rank* or *position* of item a in σ . For convenience we write $\sigma = \sigma_1 \dots \sigma_m$ and refer to the alternative at rank i as σ_i .

While complete preferences correspond to rankings, revealed preferences of a voter are typically incomplete, and may come in many forms, including incomplete rankings (with items missing from any position), top- k lists, partitioned preferences, and arbitrary pair-wise preferences, the latter being the most general case. For generality we assume that v is indeed a set of pair-wise preferences, and write $a_i \succ_v a_j$ to state that voter v prefers a_i to a_j . In-line with prior work, we assume that preferences of each individual voter are acyclic, and that the semantics of v is that of its transitive closure, denoted $tc(v)$. A voter’s preferences correspond to a complete ranking if $|tc(v)| = \binom{m}{2}$.

The Mallows model. We assume that voter preferences follow the Mallows- ϕ model [16], perhaps the most popular ranking model, and one that has been receiving significant recent attention [2, 3, 4, 15]. The model is parametrized by the modal (or reference) ranking σ and the spread (or dispersion) parameter $\phi \in (0, 1]$, and expresses the probability of observing ranking r as:

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

Here, $d(r, \sigma)$ is the distance between r and σ . The most commonly used distance is the Kendall- τ distance (also known as the bubblesort distance), which measures the number of pairs of alternatives that are in the opposite relative order in r and in σ . The normalization constant $Z(\phi)$ is given by:

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

Observe that when ϕ is close to 0, the probability mass is concentrated around σ , while for $\phi = 1$, each permutation has an equal probability.

The Mallows model was described in [16] by a *rejection sampling procedure*, which gives rise to a probability distribution over the rankings as per Equation 1. Unfortunately this procedure is very inefficient because of the high probability of rejection, especially as $\phi \rightarrow 1$.

RIM for Mallows sampling. Doignon et. al [5] introduced an efficient sampling procedure for a large class of subset choice models, of which the Mallows model is a special case. This procedure, called the *repeated insertion model* (RIM), starts with an empty ranking r and inserts items σ_1 through σ_m into r , in the order in which they appear in the modal ranking σ . Consider the contents of r after round $i - 1$. Since items are inserted into r in the order in which they appear in σ , r will contain items $\sigma_1, \dots, \sigma_{i-1}$. When RIM is used for the Mallows model, then, in round i , the probability of inserting σ_i into position $j \in [1, i]$ in r is:

$$P(\sigma_i \rightarrow r_j | \sigma, \phi) = \frac{\phi^{i-j}}{1 + \phi + \phi^2 + \dots + \phi^{i-1}} \quad (3)$$

Sampling from the Mallows posterior. RIM is an efficient sampler but it does not provide a way to sample from the Mallows posterior, which is important when the goal is, e.g., to learn the model from different kinds of training data, including complete rankings, but also top- k rankings, partitioned preferences, and pair-wise preferences. For a set of pair-wise preferences v , let us denote by $\Omega(v)$ the completions of v , i.e., the set of complete rankings that are consistent with v . The Mallows posterior is given by:

$$P(r|\sigma, \phi, v) = \frac{\phi^{d(r, \sigma)}}{\sum_{r' \in \Omega(v)} \phi^{d(r', \sigma)}} \mathbf{1}[r \in \Omega(v)] \quad (4)$$

Computing the normalization constant in the denominator of Equation 4 requires summing over an intractable number of rankings, which is to say that exact sampling from the Mallows model in presence of arbitrary pair-wise evidence is intractable. Lu and Boutilier showed this formally in [15], while also showing that efficient exact sampling is possible for restricted classes of evidence (e.g., partitioned preferences). Further, they proposed AMP, an efficient approximate sampler for arbitrary pairwise preferences. We build on AMP in our work and describe it here.

Algorithm 1 AMP Approximate Mallows Posterior

Require: Modal ranking σ , spread ϕ , preference set v .

```

1:  $\sigma_1 \rightarrow r_1$ 
2: for  $i = 2..m$  do
3:    $low = \text{calcLow}(r, tc(v), \sigma_i)$ 
4:    $high = \text{calcHigh}(r, tc(v), \sigma_i)$ 
5:   for  $j = low..high$  do
6:      $\tilde{P}(\sigma_i \rightarrow r_j) = \frac{\phi^{i-j}}{\sum_{k \in [low, high]} \phi^{i-k}}$ 
7:   end for
8:   sample rank position  $j$  from  $\tilde{P}(\sigma_i \rightarrow r)$ 
9:    $\sigma_i \rightarrow r_j$ 
10: end for
11: return  $r$ 

```

Similarly to RIM, AMP [15] generates a ranking r by inserting items σ_1 through σ_m in sequence, following their order in the modal ranking σ . At each step i , AMP considers the partial ranking r of length $i - 1$ computed so far, and computes an estimate of the probability for item σ_i to be inserted into position $j \leq i$, denoted $\tilde{P}(\sigma_i \rightarrow r_j)$. With no evidence, i.e., when $v = \emptyset$, σ_i has a non-zero probability of assignment to any position $j \in [1, i]$. However, with evidence, the range of legal positions for σ_i is restricted to some $[low, high]$, with $low \geq 0$ and $high \leq i$.

To understand the calculation in lines 3 and 4 of Algorithm 1, consider $v = \{x_1 \succ y_1, \dots, x_l \succ y_l\}$ and $\sigma = \sigma_1, \dots, \sigma_m$. Intuitively, to be consistent with v , σ_i cannot be assigned to any position that would place it ahead of some item x that is already in r and is preferred to σ_i according to v . We denote the set of such items $X = \{x | x \succ_r \sigma_i \wedge x \succ_\sigma \sigma_i\}$. Similarly, σ_i cannot be placed behind any item y that is already in r and to which it is preferred according to v , a set we denote $Y = \{y | \sigma_i \succ_r y \wedge y \succ_\sigma \sigma_i\}$. Recall that $\text{rank}_r(x)$ refers the rank of item x in r . Then, $low = \text{MAX}_{x \in X} \text{rank}_r(x)$ if $X \neq \emptyset$ and 1 otherwise, and $high = \text{MIN}_{y \in Y} \text{rank}_r(y)$ if $Y \neq \emptyset$, and i otherwise.

AMP is an approximate sampler because of the way the normalization constant is computed in line 6 of Algorithm 1.

At round i , this computation does not account for any evidence that involves items $\sigma_{i+1} \dots \sigma_m$, and so may overestimate the number of legal completions of r .

3. OUR APPROACH

3.1 Single Mallows, complete rankings

To start, let us suppose that a sample of voter preferences V is given, and that it consists of complete rankings, all drawn from a single Mallows model, to which we refer by the induced probability distribution $P(r|\sigma, \phi)$. How can we use V to estimate parameters of the model, σ and ϕ ?

Estimating modal ranking σ . It is argued in [15] that the Kemeny ranking — a ranking that has the lowest average pair-wise distance from all rankings in the sample — is the maximum likelihood estimator (MLE) for σ in the Mallows model. While identifying the optimal Kemeny ranking is NP-hard, an approximate solution can be identified very effectively in practice using local Kemenization [6]. This procedure computes σ by starting with some ranking r and then greedily swapping adjacent pairs of items as long as the average distance to rankings in the sample decreases. We implement local Kemenization using the pair-wise preference matrix, an efficient data structure that does not require to recompute distances at every step. How quickly local Kemenization identifies a local minimum depends on a good starting candidate r . In our implementation we initialize r to the ranking that occurs most frequently in the sample.

Estimating spread ϕ . Having computed an estimate for σ , we estimate ϕ using the method of moments. In a sample of rankings over m items drawn from $P(r|\sigma, \phi)$, the expected distance of a ranking from σ is given by:

$$\mathbb{E}(D) = \sum_{d:0..(\frac{m}{2})} d \cdot C_d^m \cdot \frac{\phi^d}{Z(\phi)} \quad (5)$$

Here, d is the distance of some ranking from σ , which ranges from 0 to $(\frac{m}{2})$ for Kendall- τ distance; C_d^m is the number of rankings over m items at distance d from σ ; $\frac{\phi^d}{Z(\phi)}$ is the probability of each such ranking according to the Mallows model. Note that C_d^m is the multiplier of the term ϕ^d of $Z(\phi)$ (Equation 2, when the polynomial is expanded).

Given a sample of rankings R , we can compute the mean over $r \in R$ of observed distances $d(r, \sigma)$. Equating the expression in Equation 5 with the computed mean of distances gives a polynomial over ϕ of degree m^2 . We solve this polynomial for $\phi \in [0, 1]$ using numerical methods, thus estimating the value of the spread parameter ϕ .

3.2 Single Mallows, pair-wise preferences

We now take the more realistic assumption that voter preferences V are given in the form of arbitrary pair-wise preferences. Each $v \in V$ is generated by first drawing a complete ranking r from $P(r|\sigma, \phi)$, computing its transitive closure $tc(r)$ and recording it as pairs (x, y) , where $x \succ_{tc(r)} y$, and then removing some of the item pairs. Our techniques generalize to the case where probabilities of removing a pair from $tc(v)$ (or an item from r), are specified at the level of a pair (resp. item). However, for ease of exposition, we will assume that pairs of items are missing from $tc(r)$ uniformly at random, with some specified probability that is the same for all pairs of items and all voters.

Estimating modal ranking σ . We use local Kemenization to estimate σ for samples of arbitrary pair-wise preferences. The subtlety that arises for incomplete preferences is that the computation is now based on dissimilarity between a ranking r and a set of preference pairs v , computed as:

$$d(v, r) = \sum_{i < j \leq m} \mathbf{1}[r_j \succ r_i \in tc(v)] \quad (6)$$

With this modification, the computation is similar to that for complete rankings (Section 3.1).

Estimating spread ϕ . To estimate ϕ from V , we use an expectation-maximization (EM) framework presented in Algorithm 2. At each EM round, AMP is invoked to generate a completion for each $v \in V$. Then an estimate of ϕ is computed based on the corresponding completed sample R , and is passed to the next EM round.

Algorithm 2 EM-AMP Estimating ϕ with AMP

Require: Modal ranking σ , initial value ϕ_0 , preference workload V , convergence threshold θ , maximum EM rounds $maxRounds$.

- 1: run until convergence or until maximum number of EM rounds is reached
 - 2: **for** $i = 1..maxRounds$ **do**
 - 3: $R = \emptyset$
 - 4: **for** $v \in V$ **do**
 - 5: $R \leftarrow \text{AMP}(\sigma, \phi_{i-1}, v)$
 - 6: **end for**
 - 7: $\hat{\phi}_i = \text{estimateSpread}(R, \sigma)$
 - 8: $\tilde{\phi} = \phi_i$
 - 9: **if** $\phi_i - \phi_{i-1} \leq \theta$ **then**
 - 10: **break**
 - 11: **end if**
 - 12: **end for**
 - 13: **return** $\tilde{\phi}$
-

We now propose an improvement, called AMP \times , which is based on the observation that, since an entire *workload* V of incomplete preferences is available as input to the EM framework, it can be used to provide frequency-based estimates of the necessary conditional probabilities during sampling. The idea is to estimate insertion probabilities by looking at the relevant portion of V , and to use this information instead of, or in combination with, the estimates as computed by AMP, on line 6 of Algorithm 1.

Let us denote by $v^{(j)}$ the *projection* of v onto the first $j \leq m$ items of σ : $\sigma_1, \dots, \sigma_j$. Formally, $v^{(j)} = \{x \succ y \mid x, y \in \{\sigma_1, \dots, \sigma_j\}\}$. For a preference workload $V = \{v_1, \dots, v_n\}$, we will use $V^{(j)}$ to represent $\{v_1^{(j)}, \dots, v_n^{(j)}\}$.

We now present Algorithm 3 that computes frequency-based estimates of the probability that item σ_i is inserted into positions $j = 1 \dots i$. Specifically, it computes an upper-triangular matrix, in which each row corresponds to item σ_i , and each column $j \leq i$ estimates the frequency with which σ_i is inserted into position j during sampling.

We are now ready to present Algorithm 4 that combines AMP and frequency-based estimation for approximate sampling from the Mallows posterior. The difference between AMP (Algorithm 1) and AMP \times (Algorithm 4) is that AMP \times uses an additional source of information to estimate the probability of inserting σ_i into position j in r . This information is the frequency-based estimate in IPM. The relative

Algorithm 3 IPM Insertion Probability Matrix Estimation

Require: Modal ranking σ , preference workload V .

```
1:  $T = \emptyset$ 
2: for  $v \in V$  do
3:    $T \leftarrow tc(v)$ 
4: end for
5: initialize  $M = \text{zeros}(m \times m)$ 
6: for  $i = 2..m$  do
7:   project  $T$  onto  $\sigma_1, \dots, \sigma_i$ , keep only full rankings
8:    $R = \{r \in T^{(i)} \mid isRanking(r)\}$ 
9:   for  $j = 1..i$  do
10:     $M[i, j] = |\{r \in R \mid rank_r(\sigma_i) = j\}|$ 
11:   end for
12: end for
13: return  $M$ 
```

proportion of AMP and frequency-based estimation is calculated using parameter α (lines 6-8 of Algorithm 4). We use $\alpha = 0.1$ in our experiments, because this value achieves high accuracy for a range of values of ϕ . We will show how we tuned α in Section 4.

Algorithm 4 AMPx Approximate Mallows Posterior

Require: Modal ranking σ , spread ϕ , pairwise preference set v , insertion probability matrix M , weight α .

```
1:  $\sigma_1 \rightarrow r_1$ 
2: for  $i = 2..m$  do
3:    $low = \text{calcLow}(r, tc(v), \sigma_i)$ 
4:    $high = \text{calcHigh}(r, tc(v), \sigma_i)$ 
5:    $count = \sum_{j:low..high} M[i, j]$ 
6:    $w = \frac{count}{\alpha + count}$ 
7:   for  $j = low..high$  do
8:      $\tilde{P}_{AMP}(\sigma_i \rightarrow r_j) = \frac{\phi^{i-j}}{\sum_{k \in \{low, high\}} \phi^{i-k}}$ 
9:      $\tilde{P}(\sigma_i \rightarrow r_j) = w \cdot \frac{M[i, j]}{count} + (1 - w) \cdot \tilde{P}_{AMP}(\sigma_i \rightarrow r_j)$ 
10:   end for
11:   sample rank position  $j$  from  $\tilde{P}(\sigma_i \rightarrow r)$ 
12: end for
13: return  $r$ 
```

AMPx is used in scope of an EM-based algorithm for estimating ϕ , presented in Algorithm 5. We propose multiple variants of this algorithm, which differ in whether, and how, M is maintained during EM. The basic variant, called AMPx, computes $M = \text{IPM}(\sigma, V)$ from the initial sample V (line 1), and uses it on each subsequent round of EM. The second variant, called AMPx-I (*iterative*), recomputes M at the end of each EM round, using the latest set of completed rankings R (line 14 of Algorithm 5). The third variant, called AMPx-D (*dynamic*) updates M during the EM round, incorporating completions r of preferences v as they are derived (line 7 of Algorithm 5). AMPx-I and AMPx-D can be combined and give rise to a variant we call AMPx-DI. The final variant, AMPx-S (*streaming*), is both dynamic and iterative, and, furthermore, it incorporates into M only the latest completion of a particular v . At each EM round, completion r for v , remove the contribution of v 's previous completion from M , and update M with information from r . We will show in Section 4 that AMPx-S is both significantly more accurate than AMP alone, and is also more efficient, achieving faster convergence of EM.

Algorithm 5 EM-AMPx Estimating ϕ with AMPx

Require: Modal ranking σ , initial value ϕ_0 , preference workload V , convergence threshold θ , maximum EM rounds $maxRounds$, weight α .

```
1:  $M = \text{IPM}(\sigma, V)$ 
2: for  $i = 1..maxRounds$  do
3:    $R = \emptyset$ 
4:   for  $v \in V$  do
5:      $r = \text{AMPx}(\sigma, \phi_{i-1}, v, M)$ 
6:      $R \leftarrow r$ 
7:     [AMPx-D, AMPx-S: update  $M$  with  $r$ ]
8:   end for
9:    $\phi_i = \text{estimateSpread}(R, \sigma)$ 
10:   $\tilde{\phi} = \phi_i$ 
11:  if  $\phi_i - \phi_{i-1} \leq \theta$  then
12:    break
13:  end if
14:  [AMPx-I:  $M = \text{IPM}(\sigma, R)$ ]
15: end for
16: return  $\tilde{\phi}$ 
```

3.3 Learning parameters of the mixture

Recall that our goal is to describe preferences of a diverse user population. While we assume here that each user's preferences are drawn from one particular Mallows model, the mapping between a user and a model is, of course, unknown. Further, it is unknown how many models are present in the mixture, and in what relative proportion.

The Mallows mixture distribution with K components is parametrized by mixing proportions $\pi = (\pi^{(1)}, \dots, \pi^{(K)})$, modal rankings $\sigma = (\sigma^{(1)}, \dots, \sigma^{(K)})$ and dispersion parameters $\phi = (\phi^{(1)}, \dots, \phi^{(K)})$. A common approach is to learn a mixture model for a given K using EM. The process is repeated for multiple values of K , and a mixture model is chosen that maximizes the log-likelihood of the sample [15].

Affinity propagation. Rather than attempting to fit a mixture model for varying values of K , we use *affinity propagation* [7], a state-of-the-art clustering method based on message passing, to identify an appropriate number of Mallows models (clusters) and their respective reference rankings, called exemplars here. We now briefly explain the affinity propagation algorithm. We use Aprox¹, a Java parallelized library implementation of affinity propagation.

Consider again a preference workload V of size n (n is the number of voters). AP takes as input an $n \times n$ matrix S , where $S[i, j]$ represents the *similarity* between v_i and v_j , computed in our case as the number of pairs that are in common to v_i and v_j : $S[i, j] = S[j, i] = |tc(v_i) \cap tc(v_j)|$. Similarity represents the strength of endorsement that v_i receives from v_j to become its cluster centroid. Diagonal entries $S[i, i]$ contain the a priori *suitability* of v_i to serve as an exemplar. We initially set all $S[i, i]$ uniformly to the minimum observed pair-wise similarity.

AP runs in iterations until convergence. At each iteration, each data point returns the identity of some point x , which it elects as its exemplar. All data points that return the same x as their exemplar are considered to belong to the same cluster. Proportions π are determined based on the number of points belonging to each cluster. Once the num-

¹<http://www.apro.u-psud.fr/>

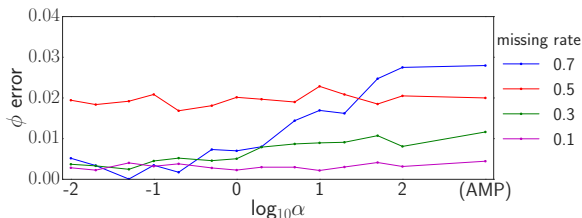


Figure 1: $\phi = 0.2$

ber of clusters and their members are settled, we estimate parameters σ and ϕ of the individual Mallows models using techniques of Sections 3.1 and 3.2.

Tuning the number of clusters. While we do not specify the number of clusters we wish to return, we do bound the number of clusters that we expect. If more than the expected number of clusters are produced, we run another round of AP over the cluster centroids. This process continues recursively, until either the number of clusters no longer changes, or it falls below the required maximum.

4. EXPERIMENTAL EVALUATION

Data and Methodology. Our experimental evaluation was conducted with synthetic and real datasets. Experiments that investigate accuracy of reconstruction of parameters of a single Mallows were executed on synthetic data, because we need information about the ground truth to evaluate accuracy. We used two real datasets, Sushi [13] and CrowdRank [18], to evaluate performance of affinity propagation, and the over-all quality of reconstruction.

In all experiments, V is generated with full rankings, and then items were removed uniformly at random, with probability corresponding to the missing rate. In all experiments we split the data into training (70%) and test (30%), train the model on the training data, and report accuracy on the test dataset. All results are averages over 15 random splits.

Tuning the α parameter. In this experiment we focus on determining an appropriate value for the α parameter, which specifies the relative contributions of AMP and of frequency-based posterior probability estimation in Algorithm 4. The basic AMPx variant is used in this experiment. We fix $m = 50$, $|V| = m^2$, vary $\phi \in \{0.2, 0.5, 0.8\}$, and set σ randomly. To not give an a priori advantage to particular values of ϕ , we start with an initial estimate ϕ_0 such that $|\phi_0 - \phi| = 0.3$.

Figure 1 presents observed error rates as a function of α for $\phi = 0.2$. Trends for $\phi = 0.5$ and $\phi = 0.8$ were similar, although the differences in accuracy for different values of α were less pronounced. We observe that accuracy is more sensitive to α for lower values of ϕ and for higher missing rates; $\alpha = 0.1$ gives good accuracy across the board, and we use this value in the remainder of our experiments.

Comparison between AMPx variants. In the next experiment we compare the accuracy of AMP, AMPx and its variants for reconstruction of parameters of a single Mallows. We vary $\phi \in \{0.2, 0.5, 0.8\}$, number of items $m \in \{25, 50, 75, 100, 125, 150\}$, missing rate in $\{0.1, 0.3, 0.5, 0.7\}$. The relative performance of AMPx variants is consistent across these settings, and we present plots for 50 items, for the highest missing rate 0.7 and for $\phi = 0.2$. This combination of missing rate and ϕ is the most challenging in terms of accuracy.

Figures 2 and 3 present the relative performance of AMP, AMPx, AMPx-I, AMPx-D, AMPx-DI, and AMPx-S for two

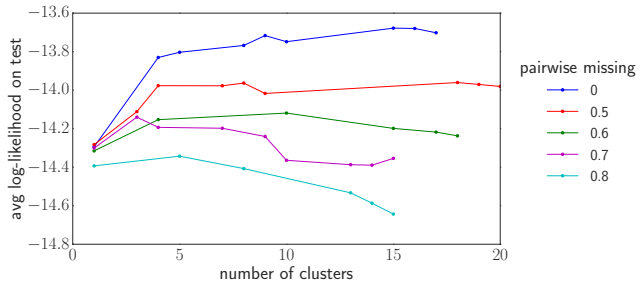


Figure 4: Log-likelihood on test data, sushi-a.

representative cases. Observe that all variants are able to reconstruct ϕ with only small errors, but that their accuracy and time to convergence differ. Of all variants, AMPx-S achieves highest accuracy, and converges after 16 iterations (on average) for the most difficult case, with $\phi = 0.2$ and missing rate 0.7. For the same case, AMPx needs about 29 iterations. We also experimented with AMPx-D (not included in plots), which showed fast convergence initially, but then failed to fully converge. Our conclusion is that dynamic variants show faster convergence, while iterative variants show lower variance. AMPx-S combines the best of both approaches. We used the Kolmogorov-Smirnov test to check whether differences in accuracy between methods are statistically significant. We found that AMPx outperforms AMP; AMPx-DI and AMPx-I both outperform AMPx (and so also AMP), and that AMPx-S outperforms all other variants, all with statistical significance ($p < 0.00001$).

Figures 2 and 3 also show that, in addition to being more accurate, AMPx-S is the over-all winner in terms of running time: it is at least as efficient as AMP in all cases, and often outperforms AMP by 30-40%.

Affinity propagation. In this experiment we used affinity propagation to reconstruct parameters of the Mallows mixture, using real datasets. Figure 4 shows the log-likelihood on the Sushi-a [13] dataset. This dataset was used for evaluation in [15], and we run our experiment at the same settings. To be consistent with [15], we apply the missing rate to pair of items in the transitive closure of each $v \in V$, rather than to individual items in the full ranking for v , before computing its transitive closure. In [15] EM was used to tune the number of clusters, and 6 clusters were found to have best log-likelihood, reported to be in the -14.3 to -13.7 range, depending on missing rate. According to Figure 4, we achieve best log-likelihood for 6 clusters, with values in the -14.4 to -13.8 range.

Importantly, affinity propagation is very efficient. The over-all time of clustering and individual model reconstruction (with AMPx-S) is between 20 and 40 seconds in our experiments, with higher running times corresponding to lower missing rates, when there is simply more data to cluster. These times can be further improved in a parallel implementation.

We also evaluated our framework with CrowdRank [18], a crowdsourced datasets of movie rankings, made up of 50 human intelligence tasks (HITs), with users ranking up to 20 movies in each HIT. We found that 28 HITs showed agreement among users, leading to 1 Mallows model each, with mean ϕ in the 0.70 to 0.95 range; 20 HITs resulted in 2 Mallows models each, with mean ϕ between 0.50 and 0.92; 2 HITs resulted in 3 Mallows models each, with ϕ ranging from 0.50 to 0.79 (mean 0.58).

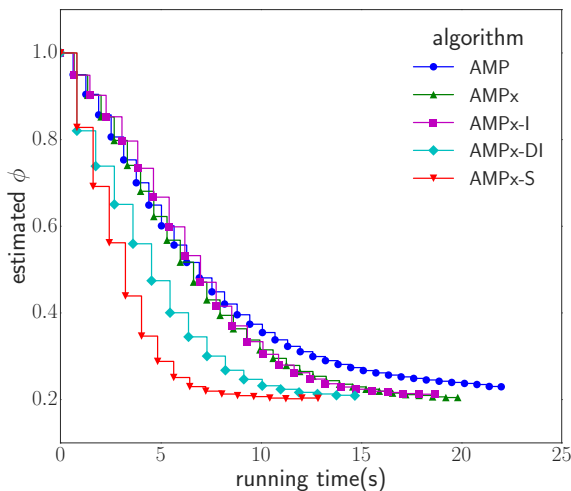


Figure 2: $\phi = 0.2$, missing rate 0.7.

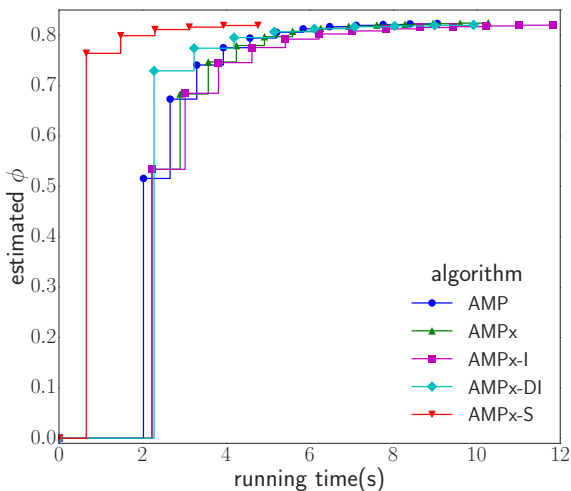


Figure 3: $\phi = 0.8$, missing rate 0.7.

5. RELATED WORK

There has been much work on methods for preference data modeling and analysis. Motivated by survey data, a variety of preference models have been developed in the statistics literature [17]. In recent years, there has been increased interest in analyzing ranked data in the machine learning [2, 3, 11, 14, 15] community. We build on the work of Lu and Boutilier [15] and develop an improved approximate Mallows posterior sampler.

In rank aggregation, a collection of input rankings is aggregated to find a ranking that intuitively represents a consensus or summary. Aggregation methods range from very simple, such as sorting items on the sum of their ranks (the Borda count), to more sophisticated, where one searches for rankings that minimize the overall distance from each ranking in the input set (Kemeny optimal aggregation). Kemeny aggregation is NP-hard [6], and several tractable relaxations [1, 6] have been proposed.

6. CONCLUSIONS

We presented a method for workload-driven learning of Mallows mixtures with pairwise preference data. We proposed AMPx, an ensemble method for approximate sampling from the Mallows posterior that combines AMP with frequency-based estimation of posterior probabilities. We experimentally demonstrated that AMPx achieves faster convergence and higher accuracy than AMP alone. Further, we showed that affinity propagation can be used to efficiently and accurately find the number of Mallows models in a mixture. All results of our work, including code, data generators, experimental methodology, and experimental results, are available at DB4Pref.com.

7. REFERENCES

- [1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *JACM*, 55(5):23, 2008.
- [2] P. Awasthi et al. Learning mixtures of ranking models. In *NIPS*.
- [3] L. M. Busse, P. Orbanz, and J. M. Buhmann. Cluster analysis of heterogeneous rank data. In *ICML*, 2007.
- [4] W. Ding, P. Ishwar, and V. Saligrama. Learning mixed membership mallows models from pairwise comparisons. *CoRR*, abs/1504.00757, 2015.
- [5] J.-P. Doignon, A. Pekeč, and M. Regenwetter. The repeated insertion model for rankings: missing link between two subset choice models. *Psychometrika*, 69(1), 2004.
- [6] C. Dwork et al. Rank aggregation methods for the web. In *WWW*, 2001.
- [7] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814), 2007.
- [8] I. C. Gormley and T. B. Murphy. A latent space model for rank data. In *ICML*, 2006.
- [9] I. C. Gormley and T. B. Murphy. A mixture of experts model for rank data with applications in election studies. *The Annals of Applied Statistics*, 2(4), 2008.
- [10] J. Huang and C. Guestrin. Learning hierarchical riffle independent groupings from rankings. In *ICML*, 2010.
- [11] J. Huang, A. Kapoor, and C. Guestrin. Efficient probabilistic inference with partial ranking queries. In *UAI*, 2011.
- [12] M. Jacob, B. Kimelfeld, and J. Stoyanovich. A system for management and analysis of preference data. *PVLDB*, 7(12), 2014.
- [13] T. Kamishima and S. Akaho. Supervised ordering by regression combined with thurstone’s model. *Artif. Intell. Rev.*, 25(3), 2006.
- [14] G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. *JMLR*, 9, 2008.
- [15] T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *JMLR*, 15(1), 2014.
- [16] C. Mallows. Non-null ranking models. I. *Biometrika*, 44, 1957.
- [17] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, 1995.
- [18] J. Stoyanovich, M. Jacob, and X. Gong. Analyzing crowd rankings. In *WebDB*, 2015.