Unsupervised Rank Aggregation with Distance-Based Models

Alexandre Klementiev KLEMENTI@UIUC.EDU
Dan Roth DANR@UIUC.EDU
Kevin Small KSMALL@UIUC.EDU
University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, IL 61801 USA

Abstract

The need to meaningfully combine sets of rankings often comes up when one deals with ranked data. Although a number of heuristic and supervised learning approaches to rank aggregation exist, they require domain knowledge or supervised ranked data, both of which are expensive to acquire. In order to address these limitations, we propose a mathematical and algorithmic framework for learning to aggregate (partial) rankings without supervision. We instantiate the framework for the cases of combining permutations and combining top-k lists, and propose a novel metric for the latter. Experiments in both scenarios demonstrate the effectiveness of the proposed formalism.

1. Introduction

Consider the scenario where each member of a panel of judges independently generates a (partial) ranking over a set of items while attempting to reproduce a true underlying ranking according to their level of expertise. This setting motivates a fundamental machine learning and information retrieval (IR) problem - the necessity to meaningfully aggregate preference rankings into a joint ranking. The IR community refers to this as *data fusion*, where a joint ranking is derived from the outputs of multiple retrieval systems. For example, in *meta-search* the aim is to aggregate Web search query results from several engines into a more accurate ranking. In many natural language processing applications, such as *machine translation*, there has been an increased interest in combining the results of multiple systems built on different principles in an effort to improve performance (Rosti et al., 2007).

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

One impediment to solving *rank aggregation* tasks is the high cost associated with acquiring full or partial preference information, making supervised approaches of limited utility. For data fusion, efforts to overcome this difficulty include applying domain specific heuristics (Shaw & Fox, 1994) or collecting such preference information indirectly (e.g. using clickthrough data (Joachims, 2002)). In order to address this limitation, we propose a general *unsupervised learning* framework for (partial) rank aggregation.

Analyzing ranked data is an extensively studied problem in statistics, information retrieval, and machine learning literature. (Mallows, 1957) introduced a distance-based model for fully ranked data and investigated its use with Kendall's and Spearman's metrics. The model was later generalized to other distance functions and for use with partially ranked data (Critchlow, 1985). (Lebanon & Lafferty, 2002) proposed a multi-parameter extension, where multiple modal rankings (e.g. expert opinions) are available and use their formalism for supervised ensemble learning; they also analyzed their model for partially ranked data (Lebanon & Lafferty, 2003).

The first key contribution of our work is the derivation of an EM-based algorithm for learning the parameters of the extended Mallows model without supervision. We instantiate the model with appropriate distance functions for two important scenarios: combining permutations and combining top-k lists. In the context of defining distances between rankings, various metrics have been proposed and analyzed (Critchlow, 1985; Estivill-Castro et al., 1993). Distances over top-k lists, i.e. rankings over the k most preferable objects, receive particular attention in the IR community (Fagin et al., 2003). (Fligner & Verducci, 1986) show that a class of distance functions between full rankings, such as Kendall's and Cayley's metrics, decompose into a sum of independent components allowing for efficient parameter estimation of the standard Mallows model.

A second key contribution of our work is the derivation

of a novel decomposable distance function for top-k lists. We show it to be a generalization of the Kendall metric and demonstrate that it can be decomposed, enabling us to estimate the parameters of the extended Mallows model efficiently.

Among recent work, (Busse et al., 2007) propose a method for clustering heterogeneous rank data based on the standard Mallows model. More directly related, many heuristics as well as a number of supervised learning approaches (Liu et al., 2007) exist for rank aggregation, although few *learn* to combine rankings without any supervision. (Klementiev et al., 2007) frame unsupervised rank aggregation as an optimization problem specifically for top-k lists, which relies on user-tuned parameters, a form of implicit supervision, whereas we describe a general unsupervised framework that can be instantiated to top-k lists in addition to other settings.

The remainder of the paper is organized as follows: section 2 formalizes distance-based ranking models and introduces relevant notation. Section 3 derives our EM-based algorithm for learning model parameters and specifies the requirements for efficient learning and inference. Section 4 instantiates the framework for two common scenarios: permutations (full rankings) and top-k lists. Section 5 experimentally demonstrates the model's effectiveness in both cases. Finally, section 6 concludes the work and gives ideas for future directions.

2. Distance-Based Ranking Models

2.1. Notation and Definitions

Let $\{x_1, \ldots, x_n\}$ be a set of objects to be ranked, i.e. assigned rank-positions $1, \ldots, n$, by a judge. We denote the resulting permutation $\pi = (\pi(1), \ldots, \pi(n))$, where $\pi(i)$ is the rank assigned to object x_i. Correspondingly, we use $\pi^{-1}(j)$ to denote the index of the object assigned to rank j.

Let $\mathcal{S}_n$ be the set of all $n!$ permutations over n items, and let $d : \mathcal{S}_n \times \mathcal{S}_n \to \mathbb{R}$ be a distance function between two permutations. We will require $d(\cdot, \cdot)$ to be a *right-invariant metric* (Diaconis & Graham, 1977): in addition to the usual properties of a metric, we will also require that the value of $d(\cdot, \cdot)$ does not depend on how the set of objects is indexed. In other words, $d(\pi, \sigma) = d(\pi\tau, \sigma\tau)\ \forall \pi, \sigma, \tau \in \mathcal{S}_n$, where $\pi\tau$ is defined by $\pi\tau(i) = \pi(\tau(i))$.

In particular, note that $d(\pi, \sigma) = d(\pi\pi^{-1}, \sigma\pi^{-1}) = d(e, \sigma\pi^{-1})$, where $e = (1, \ldots, n)$ is the identity permutation. That is, the value of d does not change if we re-index the objects such that one of the permutations becomes e and the other $\nu = \sigma\pi^{-1}$. Borrowing the notation from (Fligner & Verducci, 1986) we abbreviate $d(e, \nu)$ as $D(\nu)$. In a later section, when we define ν as a random variable, we may treat $D(\nu) = D$ as a random variable as well: whether it is a distance function or a r.v. will be clear from the context.

2.2. Mallows Models

While a large body of work on ranking models exists in statistics literature, of particular interest to us are the distance based conditional models first introduced in (Mallows, 1957). Let us give a brief review of the formalism and elucidate some of the its properties relevant to our work. The model generates a judge's rankings according to:

$$p(\pi|\theta, \sigma) = \frac{1}{Z(\theta, \sigma)} \exp(\theta\ d(\pi, \sigma)) \tag{1}$$

where $Z(\theta, \sigma) = \sum_{\pi \in \mathcal{S}_n} \exp(\theta\ d(\pi, \sigma))$ is a normalizing constant. The parameters of the model are $\theta \in \mathbb{R}$, $\theta \leq 0$ and $\sigma \in \mathcal{S}_n$, referred to as the dispersion and the location parameters, respectively. The distribution's single mode is the modal ranking σ; the probability of ranking π decreases exponentially with distance from σ. When $\theta = 0$, the distribution is uniform, and it becomes more concentrated at σ as θ decreases.

One property of (1) is that the normalizing constant $Z(\theta, \sigma)$ does not depend on σ due to the right invariance of the distance function:

$$Z(\theta, \sigma) = Z(\theta) \tag{2}$$

Let us denote the moment generating function of D under (1) as $M_{D,\theta}(t)$, and as $M_{D,0}(t)$ under the uniform distribution ($\theta = 0$). Since (1) is an exponential family,

$$M_{D,\theta}(t) = \frac{M_{D,0}(t + \theta)}{M_{D,0}(\theta)}$$

Therefore,

$$\begin{aligned} E_\theta(D) &= \frac{1}{M_{D,0}(\theta)} \frac{dM_{D,0}(t+\theta)}{dt}\bigg|_{t=0} \\ &= \frac{d \ln(M_{D,0}(t))}{dt}\bigg|_{t=\theta} \end{aligned} \tag{3}$$

(Fligner & Verducci, 1986) note that if a distance function can be expressed as $D(\pi) = \sum_{i=1}^{m} V_i(\pi)$, where

$V_i(\pi)$ are independent (with π uniformly distributed) with m.-g.f. $M_i(t)$, then $M_{D,0}(t) = \prod_{i=1}^{m} M_i(t)$. Consequently, (3) gives:

$$E_\theta(D) = \frac{d}{dt} \sum_{i=1}^{m} \ln M_i(t) \Big|_{t=\theta} \quad (4)$$

We will call such distance functions *decomposable* and will later use (4) in section 4 in order to estimate θ efficiently.

2.3. Extended Mallows Models

(Lebanon & Lafferty, 2002) propose a natural generalization of the Mallows model to the following conditional model:

$$p(\pi|\boldsymbol{\theta}, \boldsymbol{\sigma}) = \frac{1}{Z(\boldsymbol{\theta}, \boldsymbol{\sigma})} p(\pi) \exp\left(\sum_{i=1}^{K} \theta_i \; d(\pi, \sigma_i) \right) \quad (5)$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K) \in \mathcal{S}_n^K$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K$, $\boldsymbol{\theta} \leq \mathbf{0}$, $p(\pi)$ is a prior, and normalizing constant $Z(\boldsymbol{\theta}, \boldsymbol{\sigma}) = \sum_{\pi \in \mathcal{S}_n} p(\pi) \exp(\sum_{i=1}^{K} \theta_i \; d(\pi, \sigma_i))$.

The rankings σ_i may be thought of as votes of K individual judges, e.g. rankings returned by multiple search engines for a particular query in the meta-search setting. The free parameters θ_i represent the degree of expertise of the individual judges: the closer the value of θ_i to zero, the less the vote of the i-th judge affects the assignment of probability.

Under the right-invariance assumption on d, we can use property (2) to derive the following generative story underlying the extended Mallows model:

$$p(\pi, \boldsymbol{\sigma}|\boldsymbol{\theta}) = p(\pi) \prod_{i=1}^{K} p(\sigma_i|\theta_i, \pi) \quad (6)$$

That is, π is first drawn from prior $p(\pi)$. $\boldsymbol{\sigma}$ is then made up by drawing $\sigma_1 \ldots \sigma_K$ *independently* from K Mallows models $p(\sigma_i|\theta_i, \pi)$ with the *same* location parameter π.

It is straightforward to generalize both Mallows models (Critchlow, 1985), and the extended Mallows models to *partial rankings* by constructing appropriate distance functions. We will assume this more general setting in the following section.

3. Learning and Inference

In this section, we derive the general formulation of Expectation Maximization algorithm for parameter estimation of the extended Mallows models (5), and suggest a class of distance functions for which learning can be done efficiently. We then describe an inference procedure for the model.

3.1. EM Background and Notation

Let us start with a brief overview of Expectation-Maximization (Dempster et al., 1977) mostly to introduce some notation. EM is a general method of finding maximum likelihood estimate of parameters of models which depend on unobserved variables. The EM procedure iterates between:

E step: estimate the expected value of complete data log-likelihood with respect to unknown data $\mathcal{Y}$, observed data $\mathcal{X}$, and current parameter estimates θ':

$$T(\theta, \theta') \quad = \quad E[\log p(\mathcal{X}, \mathcal{Y}|\theta)|\mathcal{X}, \theta']$$

M step: choose parameters that maximize the expectation computed in the E step:

$$\theta' \leftarrow \operatorname*{argmax}_{\theta} T(\theta, \theta')$$

In our setting, the $K > 2$ experts generate votes $\boldsymbol{\sigma}$ corresponding to the unobserved true ranking π. We will see multiple instances of $\boldsymbol{\sigma}$ so the observed data we get are ranking vectors $\mathcal{X} = \{\boldsymbol{\sigma}^{(j)}\}_{j=1}^{Q}$ with the corresponding true (unobserved) rankings $\mathcal{Y} = \{\pi^{(j)}\}_{j=1}^{Q}$.

In the meta-search example, $\sigma_i^{(j)}$ is the ranking of the i-th (of the total of K) search engine for the j-th (of the total of Q) query. The (unknown) true ranking corresponding to the j-th query is denoted as $\pi^{(j)}$.

3.2. EM Derivation

We now use the generative story (6) to derive the following propositions (proofs omitted due to space constraints):

Proposition 1. *The expected value of the complete data log-likelihood under* (5) *is:*

$$T(\boldsymbol{\theta}, \boldsymbol{\theta'}) = \sum_{(\pi^{(1)}, \ldots, \pi^{(Q)}) \in \mathcal{S}_n^Q} \mathcal{L}_{\boldsymbol{\theta}} \; \mathcal{U}_{\boldsymbol{\theta'}} \quad (7)$$

where the complete data log-likelihood $\mathcal{L}_{\boldsymbol{\theta}}$ *is:*

$$\mathcal{L}_{\boldsymbol{\theta}} = \sum_{j=1}^{Q} \log p(\pi^{(j)}) - Q\sum_{i=1}^{K} \log Z(\theta_i) + \sum_{j=1}^{Q}\sum_{i=1}^{K} \theta_i \; d(\pi^{(j)}, \sigma_i^{(j)})$$

and the marginal distribution of the unobserved data $\mathcal{U}_{\boldsymbol{\theta'}}$ *is:*

$$\mathcal{U}_{\boldsymbol{\theta'}} = \prod_{j=1}^{Q} p\left(\pi^{(j)} | \boldsymbol{\theta'}, \boldsymbol{\sigma}^{(j)}\right)$$

Proposition 2. *$T(\boldsymbol{\theta}, \boldsymbol{\theta'})$ is maximized by $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ such that:*

$$E_{\theta_i}(D) = \sum_{\substack{(\pi^{(1)},\ldots,\pi^{(Q)}) \\ \in \mathcal{S}_n^Q}} \left(\frac{1}{Q}\sum_{q=1}^{Q} d(\pi^{(q)}, \sigma_i^{(q)})\right) \mathcal{U}_{\boldsymbol{\theta'}} \quad (8)$$

That is, on each iteration of EM, we need to evaluate the right-hand side (RHS) of (8) and solve the LHS for θ_i for each of the K components.

3.3. Model Learning and Inference

At first, both evaluating the RHS of (8) and solving the LHS for θ_i seem quite expensive ($> n!$). While true in general, we can make the learning tractable for a certain type of distance functions.

In particular, if a distance function can be decomposed into a sum of independent components under the uniform distribution of π (see section 2.2), property (4) may enable us to make the estimation of the LHS efficient. In Section 4, we show two examples of such distance functions (for permutations and top-k lists).

In order to estimate the RHS, we use the Metropolis algorithm (Hastings, 1970) to sample from (5). The chain proceeds as follows: denoting the most recent value sampled as π_t, two indices $i, j \in \{1, \ldots, n\}$ are chosen at random and the objects $\pi_t^{-1}(i)$ and $\pi_t^{-1}(j)$ are transposed forming π'_t. If $a = p(\pi'_t|\boldsymbol{\theta}, \boldsymbol{\sigma})/p(\pi_t|\boldsymbol{\theta}, \boldsymbol{\sigma}) \geq 1$ the chain moves to π'_t. If $a < 1$, the chain moves to π'_t with probability a; otherwise, it stays at π_t. (Diaconis & Saloff-Coste, 1998) show quick convergence for Mallows model with Cayley's distance. While no convergence results are known for the extended Mallows model with arbitrary distance, we found experimentally that the MC chain converges rapidly with the two distance functions used in this work ($10n$ steps in experiments of Section 5).

As the chain proceeds, we update the distance value with the incremental change due to a single transposition, instead of recomputing it from scratch, resulting in substantial savings in computation.

Alternatively, we also found (Section 5.1) that a combination of rankings σ_i weighted by $\exp(-\theta_i)$ provides a reasonable and quick estimate for evaluating the RHS.

Sampling or the suggested alternative RHS estimation used during training is also used for model inference.

4. Model Application

Overcoming the remaining hurdle (the LHS estimation) in learning the model efficiently depends on the definition of a distance function. We now consider two particular types of (partial) rankings: permutations, and top-k lists. The latter is the case when each judge specifies a ranking over k most preferable objects out of n. For instance, a top-10 list may be associated with the 10 items on the first page of results returned by a web search engine. For both permutations and top-k lists, we show distance functions which satisfy the decomposability property (Section 2.2), which, in turn, allows us to estimate the LHS of (8) efficiently.

4.1. Combining Permutations

Kendall's tau distance (Kendall, 1938) between permutations π and σ is a right-invariant metric defined as the minimum number of pairwise adjacent transpositions needed to turn one permutation into the other. Assuming that one of the permutations, say σ, is the identity permutation e (we can always turn one of the permutations into e by re-indexing the objects without changing the value of the distance, see Section 2.1), it can be written as:

$$D_K(\pi) = \sum_{i=1}^{n-1} V_i(\pi)$$

where[1] $V_i(\pi) = \sum_{j>i} I(\pi^{-1}(i) - \pi^{-1}(j))$. V_i are independent and uniform over integers $[0, n-i]$ (Feller, 1968) with m.-g.f. $M_i(t) = \frac{1}{n-i+1}\sum_{k=0}^{n-i} e^{tk}$. Following (Fligner & Verducci, 1986), equation (4) gives:

$$E_\theta(D_K) = \frac{ne^\theta}{1-e^\theta} - \sum_{j=1}^{n} \frac{je^{\theta j}}{1-e^{\theta j}} \quad (9)$$

$E_\theta(D_K)$ is monotone decreasing, so line search for θ will converge quickly.

[1] $I(x) = 1$ if $x > 0$, and 0 otherwise.

4.2. Combining Top-k Lists

We now propose an extension of the Kendall's tau distance to top-k lists, i.e. the case where π and σ indicate preferences over different (possibly, overlapping) subsets of $k \leq n$ objects.

Let us denote by F_π and F_σ the elements in π and σ respectively, noting that $|F_\pi| = |F_\sigma| = k$. We define $Z = F_\pi \cap F_\sigma$, $|Z| = z$, $P = F_\pi \setminus F_\sigma$, and $S = F_\sigma \setminus F_\pi$ (note that $|P| = |S| = k - z = r$). We treat π and σ as rankings, which for us will mean that the smallest index will indicate the top, i.e. contain the most preferred object. For notational convenience, let us now define the *augmented ranking* $\tilde{\pi}$ as π augmented with the elements of S assigned the same index $(k+1)$, one past the bottom of the ranking as shown on Figure 1 ($\tilde{\sigma}$ is defined similarly). We will slightly abuse our notation and denote $\tilde{\pi}^{-1}(k+1)$ to be the set of elements in position $(k+1)$.

Kendall's tau distance D_K is naturally extended from permutations to augmented rankings.

Definition 1. *Distance $\tilde{D}_K(\tilde{\pi}, \tilde{\sigma})$ between augmented rankings $\tilde{\pi}$ and $\tilde{\sigma}$ is the minimum number of adjacent transpositions needed to turn $\tilde{\pi}$ into $\tilde{\sigma}$.*

It can be shown that $\tilde{D}_K(\tilde{\pi}, \tilde{\sigma})$ is a right-invariant metric, thus we will again simplify the notation denoting it as $\tilde{D}_K(\tilde{\pi})$. This distance can be decomposed as:

$$\tilde{D}_K(\tilde{\pi}) = \sum_{\substack{i=1 \\ \tilde{\pi}^{-1}(i) \in Z}}^{k} \tilde{V}_i(\tilde{\pi}) + \sum_{\substack{i=1 \\ \tilde{\pi}^{-1}(i) \notin Z}}^{k} \tilde{U}_i(\tilde{\pi}) + \frac{r(r+1)}{2}$$

where

$$\begin{aligned} \tilde{V}_i(\tilde{\pi}) &= \sum_{\substack{j=i \\ \tilde{\pi}^{-1}(j) \in Z}}^{k} I(\tilde{\pi}^{-1}(i) - \tilde{\pi}^{-1}(j)) + \\ &\quad \sum_{j \in \tilde{\pi}^{-1}(k+1)} I(\tilde{\pi}^{-1}(i) - j) \\ \tilde{U}_i(\tilde{\pi}) &= \sum_{\substack{j=i \\ \tilde{\pi}^{-1}(j) \in Z}}^{k} 1 \end{aligned}$$

Decomposing $\tilde{D}_K(\tilde{\pi})$, the second term is the minimum number of adjacent transpositions necessary to bring the r elements not in Z (grey boxes on Figure 1) to the bottom of the ranking. The third term is the minimum number of adjacent transpositions needed to switch them with the elements in $\tilde{\pi}^{-1}(k+1)$, which would then appear in the correct order in the bottom r positions. Finally, the first term is the adjacent transpositions necessary to put the k elements now in the list in the natural order.

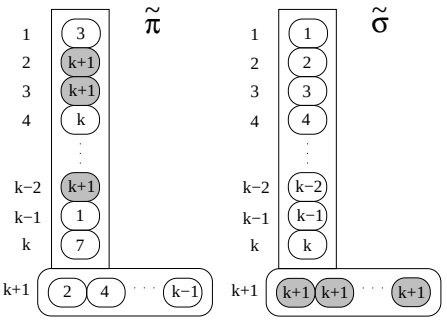

Figure 1. An example of augmented permutations $\tilde{\pi}$ (left) and identity augmented permutation $\tilde{\sigma}$ (right, in natural order). Grey boxes are objects in π but not in σ. $\tilde{D}_K(\tilde{\pi})$ is the minimum number of adjacent transpositions needed to turn $\tilde{\pi}$ into $\tilde{\sigma}$: namely, bring all grey boxes into the position $k+1$ and put the remaining k objects in their natural order.

It can be shown that the random variable summands comprising $\tilde{D}_K(\tilde{\pi})$ are independent when $\tilde{\pi}$ is uniformly distributed. Furthermore, $\tilde{V}_i$ and $\tilde{U}_j$ are uniform over integers $[0, k-i]$ and $[0, z]$, with moment generating functions $\frac{1}{k-i+1}\sum_{j=0}^{k-i} e^{tj}$ and $\frac{1}{z+1}\sum_{j=0}^{z} e^{tj}$, respectively. Assuming $z > 0$, and $r > 0$ equation (4) gives:

$$E_\theta(\tilde{D}_K) = \frac{ke^\theta}{1-e^\theta} - \sum_{j=r+1}^{k} \frac{je^{j\theta}}{1-e^{j\theta}} + \frac{r(r+1)}{2} - r(z+1)\frac{e^{\theta(z+1)}}{1-e^{\theta(z+1)}} \qquad (10)$$

If $r = 0$ (i.e. the augmented rankings are over the same objects), both the distance and the expected value reduce to the Kendall distance results. Also, if $z = 0$ (i.e. the augmented rankings have no objects in common), $\tilde{D}_K = E_\theta(\tilde{D}_K) = k(k+1)/2$, which is the smallest number of adjacent transpositions needed to move the $r = k$ objects in $\tilde{\pi}^{-1}(k+1)$ into the top k positions.

$E_\theta(\tilde{D}_K)$ is decreasing monotonically, so we can again use line search to find the value of θ. Notice that the expected value depends on the value of z (the number of common elements between the two permutations). We will compute the average value of z as we estimate the RHS of (8) and use it to solve the LHS for θ.

5. Experimental Evaluation

We demonstrate the effectiveness of our approach for permutations and top-k lists considered in Section 4.

5.1. Permutations

We first consider the scenario of aggregating permutations. For this set of experiments, the votes of $K = 10$ individual experts were produced by sampling standard Mallows models (1), with the same location parameter $\sigma^* = e$ (an identity permutation over $n = 30$ objects), and concentration parameters $\theta^*_{1,2} = -1.0$, $\theta^*_{3,..,9} = -0.05$, and $\theta^*_{10} = 0$ (the latter generating all permutations uniformly randomly). The models were sampled 10 times, resulting in $Q = 10$ lists of permutations (one for each "query"), which constituted the training data.

In addition to the sampling procedure described in Section 3.3 to estimate the RHS of (8), we also tried the following weighted Borda count approximation. For each "query" q, we took the K votes and mixed them into a single permutation $\hat{\sigma}_q$ as follows: a score for each of the n objects is computed as a weighted combination of ranks assigned to that object by individual judges. The aggregate permutation $\hat{\sigma}_q$ is obtained by sorting the objects according to their resulting scores. The weights are computed using the current values of the model parameters as $\exp(-\theta_i)$. The rationale is that the smaller the absolute value of θ_i, the lower the relative quality of the ranker, and the less it should contribute to the aggregate vote. Finally, the RHS for the i-th component is computed as the distance from its vote to $\hat{\sigma}_q$ averaged over all Q queries.

We also tried using the true permutation σ^* in place of $\hat{\sigma}_q$ to see how well the learning procedure can do.

At the end of each EM iteration, we sampled the current model (5), and computed the Kendall's tau distance between the generated permutation to the true σ^*. Figure 2 shows the model performance when sampling and the proposed approximation are used to estimate the RHS. Although the convergence is much faster with the approximation, the model trained with the sampling method achieves better performance approaching the case when the true permutation is known.

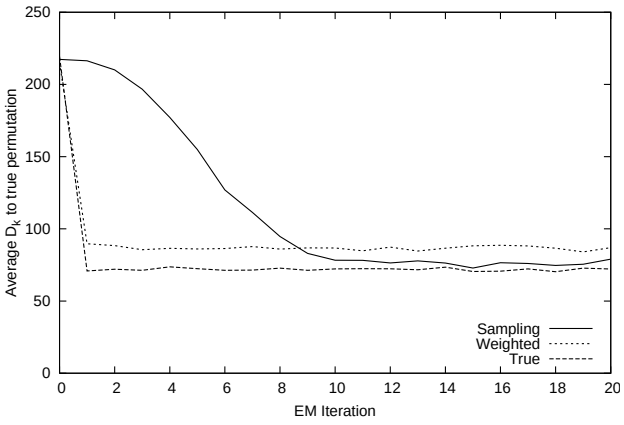

Figure 2. Permutations: learning performance of the model (averaged over 5 runs) when RHS is estimated using sampling (Sampling), the proposed weighted Borda count approximation (Weighted), or the true permutation (True). Although the convergence is much faster with the approximation, model trained with the sampling method achieves better performance.

5.2. Top-k lists

In order to estimate the model's performance in the top-k list combination scenario, we performed data fusion experiments using the data from the ad-hoc retrieval shared task of the TREC-3 conference (Harman, 1994). Our goal here is to examine the behavior of our approach as we introduce poor judges into the constituent ranker pool. In this shared task, 40 participants submitted top-1000 ranking over a large document collection for each of the 50 queries. For our experiments, we used top-100 ($k = 100$) rankings from $K = 38$ of the participants (two of the participants generated shorter rankings for some of the queries and were not used) for all $Q = 50$ queries. We replaced a specific number $K_r \in [0, K]$ of the participants with random rankers (drawing permutations of k documents from the set of documents returned by all participants for a given query uniformly randomly). We then used our algorithm to combine top-k lists from K_r random rankers and $(K - K_r)$ participants chosen at random.

We measure performance using the precision in top-$\{10, 30\}$ documents as computed by *trec_eval*[2] from the TREC conference series. As a baseline, we use $CombMNZ_{rank}$ suggested in (Klementiev et al., 2007). It is a variant of a commonly used $CombMNZ$ (Shaw & Fox, 1994). Given a query q for each document x in the collection it computes a score $N_x \times \sum_{i=1}^{K}(k - r_i(x, q))$, where $r_i(x, q)$ is the rank of the document x in the ranking returned by participant i for the query q, and N_x is the number of participants which place x in their top-k rankings. The aggregate

[2]Available at http://trec.nist.gov/

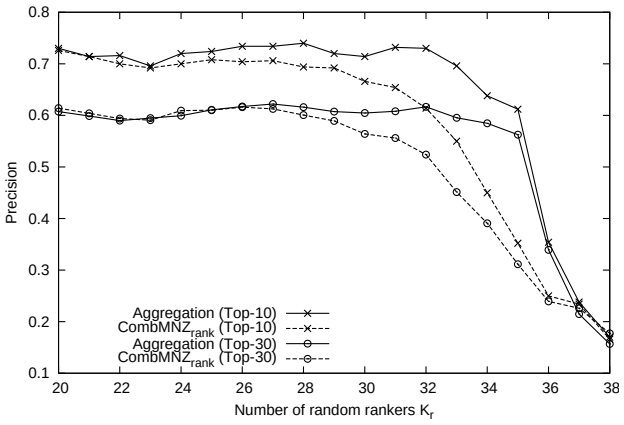

Figure 3. Top-k lists: precision of the aggregate ranker as a function of the number of random component rankers K_r in top 10 and top 30 documents. Our algorithm learns to discount the random components without supervision substantially improving over $CombMNZ_{rank}$.

ranking is obtained by sorting documents according to their scores. Intuitively, the more component rankers rank a document highly the higher it appears in the aggregate ranking.

Figure 3 shows that our algorithm learns to discount the random components *without supervision* substantially improving over the baseline as $K_r \to K$.

We also compared our results with the ULARA algorithm (Klementiev et al., 2007). These results were not included since we found ULARA to be too sensitive to user-defined parameters (an implicit form of supervision) with results varying between competitive with our model to comparable with $CombMNZ_{rank}$.

5.3. Model Dispersion Parameters

In order to demonstrate the relationship between the learned dispersion parameters of the model, $\boldsymbol{\theta}$, and the relative performance of the constituent rankers, we also conducted a meta-search experiment. First, we generated $Q = 50$ queries which result in an unambiguous most relevant document and submitted them to $K = 4$ commercial search engines. For each engine, we kept the 100 highest ranked documents (10 pages of 10 documents each) after removing duplicates, and unified URL formatting differences between engines. We measure performance with Mean Reciprocal Page Rank ($MRPR$), which we define as mean reciprocal rank of the page number on which the correct document appears.

Table 1 shows $MRPR$ of the four search engines and their corresponding model parameters. As expected, the results suggest a correlation between the magnitude of the dispersion parameters and the relative system performance, implying that their values may also be used for unsupervised search engine evaluation. Finally, our model achieves $MRPR = 0.92$ beating all of the constituent rankers.

Table 1. $MRPR$ of the four search engines and their corresponding model parameters; the results suggest a correlation between the magnitude of the dispersion parameters and the relative system performance.

	$S1$	$S2$	$S3$	$S4$
θ	-0.065	0.0	-0.066	-0.049
$MRPR$	0.86	0.43	0.82	0.78

6. Conclusions and Future Work

We propose a formal mathematical and algorithmic framework for aggregating (partial) rankings without supervision. We derive an EM-based algorithm for the extended Mallows model and show that it can be made efficient for the right-invariant decomposable distance functions. We instantiate the framework and experimentally demonstrate its effectiveness for the important cases of combining permutations and combining top-k lists. In the latter case, we introduce the notion of augmented permutation and a novel decomposable distance function for efficient learning.

A natural extension of the current work is to instantiate our framework for other types of partial rankings, as well as to cases where ranking data is not of the same type. The latter is of practical significance since often preference information available is expressed differently by different judges (e.g. top-k rankings of different lengths).

Another direction for future work is to extend the rank aggregation model to accommodate position dependence. In IR, more importance is generally given to results appearing higher in the rankings. Within our framework one may be able to design a distance function reflecting this requirement. Additionally, the quality of votes produced by individual components may depend on the rank, e.g. in the top-k scenario some rankers may be better at choosing few most relevant objects, while others may tend to have more relevant objects in the k selected but may not rank them well relative to one another. This case may be modeled by adding a dependency on rank to the dispersion parameters of the model.

In addition, this framework appears promising for a number of applications. Besides the NLP problems mentioned before, such as learning to combine output from multiple machine translation systems, one interesting setting may be *domain adaptation*. Here, the task is to adapt a hypothesis trained with ample labeled data from one input distribution to a second distribution where minimal training data is available. When the hypothesis is a trained aggregate ranker, we expect the relative expertise of its components to change and can use our approach to reweigh them accordingly.

Acknowledgments

We would like to thank Ming-Wei Chang, Sariel Har-Peled, Vivek Srikumar, and the anonymous reviewers for their valuable suggestions. This work is supported by NSF grant ITR IIS-0428472, DARPA funding under the Bootstrap Learning Program and by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

Busse, L. M., Orbanz, P., & Buhmann, J. M. (2007). Cluster analysis of heterogeneous rank data. *Proc. of the International Conference on Machine Learning (ICML)*.

Critchlow, D. E. (1985). *Metric methods for analyzing partially ranked data*, vol. 34 of *Lecture Notes in Statistics*. Springer-Verlag.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, *39*, 1–38.

Diaconis, P., & Graham, R. L. (1977). Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society*, *39*, 262–268.

Diaconis, P., & Saloff-Coste, L. (1998). What do we know about the Metropolis algorithm? *Journal of Computer and System Sciences*, *57*, 20–36.

Estivill-Castro, V., Mannila, H., & Wood, D. (1993). Right invariant metrics and measures of presortedness. *Discrete Applied Mathematics*, *42*, 1–16.

Fagin, R., Kumar, R., & Sivakumar, D. (2003). Comparing top k lists. *SIAM Journal on Discrete Mathematics*, *17*, 134–160.

Feller, W. (1968). *An introduction to probability theory and its applications*, vol. 1. John Wiley and Sons, Inc.

Fligner, M. A., & Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society*, *48*, 359–369.

Harman, D. (1994). Overview of the third Text REtrieval Conference (TREC-3).

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, *57*, 97–109.

Joachims, T. (2002). Unbiased evaluation of retrieval quality using clickthrough data. *SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, *30*, 81–93.

Klementiev, A., Roth, D., , & Small, K. (2007). An unsupervised learning algorithm for rank aggregation. *Proc. of the European Conference on Machine Learning (ECML)* (pp. 616–623).

Lebanon, G., & Lafferty, J. (2002). Cranking: Combining rankings using conditional probability models on permutations. *Proc. of the International Conference on Machine Learning (ICML)*.

Lebanon, G., & Lafferty, J. (2003). Conditional models on the ranking poset. *The Conference on Advances in Neural Information Processing Systems (NIPS)* (pp. 431–438).

Liu, Y.-T., Liu, T.-Y., Qin, T., Ma, Z.-M., & Li, H. (2007). Supervised rank aggregation. *Proc. of the International World Wide Web Conference (WWW)*.

Mallows, C. L. (1957). Non-null ranking models. *Biometrika*, *44*, 114–130.

Rosti, A.-V. I., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R., & Dorr, B. J. (2007). Combining outputs from multiple machine translation systems. *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)* (pp. 228–235).

Shaw, J. A., & Fox, E. A. (1994). Combination of multiple searches. *Text REtrieval Conference (TREC)* (pp. 243–252).