Unsupervised Rank Aggregation with Domain-Specific Expertise

Alexandre Klementiev, Dan Roth, Kevin Small, and Ivan Titov
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
{klementi,danr,ksmall,titov}@illinois.edu

Abstract

Consider the setting where a panel of judges is repeatedly asked to (partially) rank sets of objects according to given criteria, and assume that the judges' expertise depends on the objects' domain. Learning to aggregate their rankings with the goal of producing a better joint ranking is a fundamental problem in many areas of Information Retrieval and Natural Language Processing, amongst others. However, supervised ranking data is generally difficult to obtain, especially if coming from multiple domains. Therefore, we propose a framework for learning to aggregate votes of constituent rankers with domain specific expertise without supervision. We apply the learning framework to the settings of aggregating full rankings and aggregating top-k lists, demonstrating significant improvements over a domain-agnostic baseline in both cases.

1 Introduction

Consider the setting where judges are repeatedly asked to (partially) rank sets of objects, and assume that each judge tries to reproduce some true underlying ranking to the best of their ability. *Rank aggregation* aims to combine the rankings of such experts to produce a better joint ranking. Now, imagine that the judges' ability to generate rankings depends on the type of objects being ranked or the criteria they are asked to use while ranking. As a simple example, consider a group of people who are asked to rank a set of conference submissions written in English and another set written in French according to their relevance to the conference theme. One would expect that a bilingual expert in the field would produce reasonable rankings of both cases, while a judge who only speaks French and is unfamiliar with the conference topic may produce mediocre rankings (possibly, only using a set of keywords) for the French submissions and random rankings for the English set. We consider the problem of learning to aggregate these rankings by constituent judges with domain-specific expertise into a joint ranking.

The problem of aggregating rankings is ubiquitous in Information Retrieval (IR) and Natural Language Processing (NLP). In IR, for instance, *meta-search* aims to combine the outputs of multiple search engines to produce a better ranking. In machine translation (MT), aggregation of multiple systems built on different underlying principles has received considerable recent attention (e.g. [Rosti *et al.*, 2007]). In many such applications, one would expect the expertise of the constituent components to depend on the input domain. In IR, the quality of rankings produced by search engines has been shown to be query type dependent (e.g. [Geng *et al.*, 2008]): some may specialize on ranking product reviews while others on ranking scientific documents. In MT system aggregation, component systems may be trained on different corpora (e.g. multi-lingual Hansards, or technical manuals), and tend to be fragile when tested on data sampled from a different domain [Koehn and Schroeder, 2007]. Thus, the relative expertise of these systems depends on which distribution the test source language data is sampled from. Moreover, in these and many other aggregation examples, the input domain information in regards to the expertise of each judge is latent.

Supervised learning approaches to solving rank aggregation (e.g. [Liu *et al.*, 2007]) are impractical for many applications as labeled ranking data is generally very expensive to obtain. In IR, for example, heuristics or indirect methods are often employed (e.g. [Shaw and Fox, 1994; Dwork *et al.*, 2001; Joachims, 2002]) to produce a surrogate for true preference information. Needless to say, supervision for typed ranked data is even harder to obtain.

The principal contribution of this paper is a framework for learning to aggregate votes of constituent rankers with domain-specific expertise *without supervision*. Given only a set of constituent rankings, we learn an aggregation function that attempts to recreate the true ranking without labeled data. The intuition behind our approach is simple: rankers which are experts in a given domain are better at generating votes close to true rankings for that domain and thus will tend to agree with each other, whereas the non-experts will not. Given rankers' votes for a set of queries, we aim to discover patterns of rankers' agreement. Each distinct pattern corresponds to a specific (latent) domain, enabling us to deduce domain-specific expertise of each judge. The ability to identify expertise of a ranker for a specific query can potentially greatly improve accuracy of the model over simpler aggregation techniques. While our framework does not commit to any particular type of (partial) rankings, we show how to apply it to two kinds of rankings: permutations and top-k lists.

The remainder of the paper is organized as follows. Section 2 introduces relevant notation and reviews distance-based ranking models. Section 3 introduces our model for aggregating over rankers with domain-specific expertise and Section 4 derives an EM-based algorithm for learning model parameters, and describes methods to make the learning efficient. Section 5 applies the learning framework to two types of rankings: permutations (full rankings) and top-k lists. Finally, Section 6 discusses relevant work, and Section 7 concludes the work and gives ideas for future directions.

2 Distance-Based Models

While there has been a significant research effort in the statistics community with regards to analyzing and modeling ranking data (e.g. [Marden, 1995]), we are specifically interested in distance-based models beginning with the Mallows model [Mallows, 1957].

2.1 Notation and Definitions

Let us first introduce the notation we will use throughout the paper. The models presented in this work do not commit to a particular type of (partial) ranking. Therefore, we will generally use the same notation for all (partial) ranking types, and the particular type we imply should be clear from context. However, since we apply the model to two particular types of rankings, permutations (full rankings) and top-k lists, let us start with the relevant definitions.

Permutations

Let $\{x_1, x_2, \ldots, x_n\}$ be a set of objects to be ranked by a judge. A permutation π is a bijection from the set $\{1, 2, \ldots, n\}$ onto itself; we will denote by $\pi(i)$ the rank assigned to object x_i, and by $\pi^{-1}(j)$ the index of the object assigned to rank j. Let us also define e to be the identity permutation $(1, \ldots, n)$. Finally, let us denote $\mathcal{S}_n$ to be the set of all $n!$ permutations over n objects.

Let us define a distance between two permutations $d : \mathcal{S}_n \times \mathcal{S}_n \rightarrow \mathbb{R}$ and assume that, in addition to satisfying the usual metric properties, it is also *right-invariant* [Diaconis and Graham, 1977]. That is, we assume that the value of $d(\cdot, \cdot)$ does not depend on how the objects are indexed, a property natural to the applications we consider in this work. More specifically, if the objects are re-indexed by τ, the distance between two permutation over the objects does not change: $d(\pi, \sigma) = d(\pi\tau, \sigma\tau)\ \forall \pi, \sigma, \tau \in \mathcal{S}_n$, where $\pi\tau$ is defined by $\pi\tau(i) = \pi(\tau(i))$. Note that $d(\pi, \sigma) = d(\pi\pi^{-1}, \sigma\pi^{-1}) = d(e, \sigma\pi^{-1})$. That is, the value of d does not change if we re-index the objects such that one of the permutations becomes $e = (1, \ldots, n)$ and the other $\nu = \sigma\pi^{-1}$. Borrowing the notation from [Fligner and Verducci, 1986] we abbreviate $d(e, \nu)$ as $D(\nu)$. In the remainder of the paper, 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.

Examples of common right-invariant distance functions over permutations include *Kendall's tau distance*[1]:

$$d_K(\pi, \sigma) = \sum_{i=1}^{n-1} \sum_{j>i} I(\pi\sigma^{-1}(i) - \pi\sigma^{-1}(j)) \qquad (1)$$

which can be also defined as the minimum number of adjacent transpositions required to turn π into σ, and the *Spearman's footrule*:

$$d_S(\pi, \sigma) = \sum_{i=1}^{n} |\pi(i) - \sigma(i)| \qquad (2)$$

Top-k Lists

Top-k lists indicate preferences over different (possibly, overlapping) subsets of $k \leq n$ objects, where the elements not in the list are implicitly ranked below all of the list elements. They are used extensively in the IR community to represent the output of a retrieval system; a top-10 list, for instance, may represent the first page of a search engine output.

A number of distance measures over top-k lists have been proposed and studied (e.g. [Fagin *et al.*, 2003]). Of particular relevance to us will be the generalization of the Kendall's tau distance (1) to top-k lists proposed in [Klementiev *et al.*, 2008] which they defined as follows. First two top-k lists π and σ are augmented: items in σ which are not present in π are placed in the *same* position $(k + 1)$ in π, and vice versa. The *augmented Kendall's tau* distance $d_A(\pi, \sigma)$ is then defined as the minimum number of adjacent transpositions to turn the augmented π into the augmented σ. They show that the distance is right-invariant.

2.2 Mallows Models

Distance-based models for ranking data were first introduced in [Mallows, 1957], and generate judge's rankings according to:

$$p(\pi|\theta, \sigma) = \frac{1}{Z(\theta, \sigma)} \exp(\theta\, d(\pi, \sigma)) \qquad (3)$$

where $\theta \in \mathbb{R}$, $\theta \leq 0$ is the the dispersion parameter, the modal ranking $\sigma \in \mathcal{S}_n$ is the location parameter, and $Z(\theta, \sigma) = \sum_{\pi \in \mathcal{S}_n} \exp(\theta\, d(\pi, \sigma))$ is a normalizing constant. The probability of ranking π decreases exponentially with distance from the mode σ. The distribution is uniform when $\theta = 0$, and becomes sharper as θ grows more negative.

Let us note two properties of (3), which will be relevant later in the paper. Firstly, under the right invariance property of $d(\cdot, \cdot)$, the normalizing constant does not depend on σ, $Z(\theta, \sigma) = Z(\theta)$ (see, e.g. [Lebanon and Lafferty, 2002]).

Secondly, [Fligner and Verducci, 1986] note that if the distance function can be expressed as $D(\pi) = \sum_{i=1}^{m} V_i(\pi)$, where random variables $V_i(\pi)$ are independent (with π uniformly distributed), then the MLE of θ under (3), which is the solution to equation $E_\theta(D) = \overline{D}$, may be efficient to compute. [Klementiev *et al.*, 2008] refer to such distance functions as *decomposable*.

[Mallows, 1957] first investigated the model with Kendall's and Spearman's metrics on fully ranked data, and the model was later generalized to other distance functions and for use with partially ranked data [Critchlow, 1985].

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

2.3 Extended Mallows Models

Since the Mallows model was introduced, various extensions have been proposed in statistics and machine learning literature (e.g. [Fligner and Verducci, 1986; Murphy and Martin, 2003; Busse *et al.*, 2007]). Of particular interest to us is the multiple input rankings scenario proposed by [Lebanon and Lafferty, 2002] in the context of supervised learning.

Assuming that a vector of votes $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)$ from K individual judges is available, the generalized model assigns a probability to ranking π according to:

$$p(\pi|\boldsymbol{\theta}, \boldsymbol{\sigma}) = \frac{1}{Z(\boldsymbol{\theta}, \boldsymbol{\sigma})} p(\pi) \exp\left(\sum_{i=1}^{K} \theta_i \, d(\pi, \sigma_i)\right) \quad (4)$$

where $\boldsymbol{\sigma} \in \mathcal{S}_n^K$, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K$, $\boldsymbol{\theta} \leq \mathbf{0}$, $p(\pi)$ 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 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 property assumption on $d(\cdot, \cdot)$, the model has the following associated generative story:

$$p(\pi, \boldsymbol{\sigma}|\boldsymbol{\theta}) = p(\pi) \prod_{i=1}^{K} p(\sigma_i|\theta_i, \pi) \quad (5)$$

That is, π is first drawn from prior $p(\pi)$, and the votes of the K individual judges are produced by drawing *independently* from K Mallows models $p(\sigma_i|\theta_i, \pi)$ with the *same* location parameter π.

[Klementiev *et al.*, 2008] propose an Expectation-Maximization (EM) [Dempster *et al.*, 1977] based algorithm for learning the parameters of the extended Mallows model from Q vectors of votes $\{\boldsymbol{\sigma}^{(j)}\}_{j=1}^{Q}$. Their intuition is that better rankers tend to exhibit agreement more than poor ones (assumed not to collude), which was supported empirically.

In the meta-search setting, for example, the observed data is comprised of the rankings $\boldsymbol{\sigma}^{(j)}$ produced by K search engines for each of the Q queries, and the goal is to infer the joint ranking π.

3 Distance-Based Models with Domain Expertise

Up to this point, we have only considered *type agnostic* models. These models assume that the expertise of the constituent rankers do not depend on the input data domain (e.g. types of queries in the meta-search setting, or the distributions of the test sentences in the MT aggregation setting), or equivalently, that all input comes from the same domain. In the following discussion we will correspondingly refer to input data domains as *types*.

As we have argued, in practical applications, it is more natural to model the data generation process such that domain-specific expertise of the rankers is accounted for explicitly. We now proceed to the task we set out to investigate: the case of aggregating votes of rankers with domain-specific expertise. We propose a mixture of the extended distance-based models (4) as a means to formalize and model this setting.

3.1 Mixture Model

We begin by augmenting the generative story (5) to include the notion of types. First, a type t is selected from T types with probability α_t. Then, the location parameter (true ranking) π is drawn uniformly, and the votes of individual experts are drawn *independently* from K Mallows models $p(\sigma_i|\theta_{t,i}, \pi)$ with the *same* location parameter π. That is,

$$p(\pi, \boldsymbol{\sigma}, t|\boldsymbol{\theta}, \boldsymbol{\alpha}) \propto \alpha_t \prod_{i=1}^{K} p(\sigma_i|\pi, \theta_{t,i}) \quad (6)$$

The right-invariance property of the distance function can be used to derive the corresponding conditional model:

$$p(\pi, t|\boldsymbol{\sigma}, \boldsymbol{\theta}, \boldsymbol{\alpha}) = \alpha_t \frac{\exp\left(\sum_{i=1}^{K} \theta_{t,i} \, d(\pi, \sigma_i)\right)}{Z(\boldsymbol{\theta}, \boldsymbol{\sigma})} \quad (7)$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K) \in \mathcal{S}_n^K$, $\boldsymbol{\theta} \in \mathbb{R}^{T \times K}$, $\boldsymbol{\theta} \leq \mathbf{0}$, and normalizing constant $Z(\boldsymbol{\theta}, \boldsymbol{\sigma}) = \sum_{t=1}^{T} \sum_{\pi \in \mathcal{S}_n} \alpha_t \exp(\sum_{i=1}^{K} \theta_{t,i} \, d(\pi, \sigma_i))$.

The free model parameters are a $T \times K$ matrix $\boldsymbol{\theta}$, where $\theta_{t,i}$ represent the degree of expertise of the judge i for type t, and T mixture weights $\boldsymbol{\alpha}$. Note that this model is more expressive than the type agnostic model which has a single free parameter θ_i to model the expertise of each judge.

4 Learning and Inference

We now derive an EM-based algorithm for learning the free model parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$, and propose methods to make both learning and inference efficient.

4.1 Learning

Let us denote the estimates of parameters in the previous EM iteration with $\boldsymbol{\alpha'}$ and $\boldsymbol{\theta'}$. In our setting, the examples we observe are vectors of rankings $\{\boldsymbol{\sigma}^{(j)}\}_{j=1}^{Q}$, where $\sigma_i^{(j)}$ is the ranking produced by i-th ranker for example j. The unobserved data are the corresponding true rankings with the associated types: $\{(t^{(j)}, \pi^{(j)})\}_{j=1}^{Q}$.

In order to make the learning process more stable we use a symmetric Dirichlet prior $Dir(\beta)$ on the topic distribution $\boldsymbol{\alpha}$. Now, following the generative story in Section 3.1 and taking into account the prior defined on $\boldsymbol{\alpha}$, we derive the M step (proofs are omitted due to space restrictions):

Proposition 1 *The expected value of the complete data log-likelihood under* (7) *is maximized by* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\theta}$ *such that:*

$$\alpha_t = \frac{1}{T\beta + Q} \left(\beta + \sum_{j=1}^{Q} \sum_{\pi^{(j)} \in \mathcal{S}_n} p(\pi^{(j)}, t|\boldsymbol{\sigma}^{(j)}, \boldsymbol{\theta'}, \boldsymbol{\alpha'})\right) \quad (8)$$

$$E_{\theta_{t,i}}(D) = \frac{1}{\alpha_t Q} \sum_{j=1}^{Q} \sum_{\pi^{(j)} \in \mathcal{S}_n} d(\pi^{(j)}, \sigma_i^{(j)}) p(\pi^{(j)}, t|\boldsymbol{\sigma}^{(j)}, \boldsymbol{\theta'}, \boldsymbol{\alpha'}) \quad (9)$$

Thus, on each iteration of EM, (8) is used to update $\boldsymbol{\alpha}$ (T updates) and (9) are used to update $\boldsymbol{\theta}$ parameters ($T \times K$ updates).

Evaluating (8), estimating the right-hand side of (9), and then solving the left hand side for $\theta_{t,i}$ directly are all computationally intractable. However, learning can be made efficient when particular properties of the distance function discussed in Section 4.1 are satisfied.

The ideas presented so far can be applied to any kind of (partial) rankings. However, in order to propose efficient alternatives to direct estimation of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$, we need to commit to specific ranking types. Let us consider the cases of combining permutations and combining top-k lists as two examples, and address the three problems individually.

Estimating the right-hand side of (9)

In order to estimate the right-hand side of (9) we need to obtain samples from the model (7). For high dimensional multimodal distributions such as (7), standard sampling methods (e.g. Metropolis-Hastings [Hastings, 1970]) do not converge in a reasonable amount of time. Annealing methods [Neal, 1996] are still computationally expensive, and additionally require careful tuning of free parameters (i.e. annealing schedule). Therefore, we use the fast approximate sampling algorithm described below.

We start by obtaining a sample from each mixture component t. This is done using the Metropolis-Hastings algorithm applied to the extended Mallows model (4). The chain is constructed as follows: denoting the most recent value sampled as $\tau_{(t)}$, two indices $i, j \in \{1, \ldots, n\}$ are chosen at random and the objects $\tau_{(t)}^{-1}(i)$ and $\tau_{(t)}^{-1}(j)$ are transposed forming a new permutation $\tilde{\tau}_{(t)}$. If $a = p(\tilde{\tau}_{(t)}, t|\boldsymbol{\sigma}, \boldsymbol{\theta'}, \boldsymbol{\alpha'})/p(\tau_{(t)}, t|\boldsymbol{\sigma}, \boldsymbol{\theta'}, \boldsymbol{\alpha'}) \geq 1$ the chain moves to $\tilde{\tau}_{(t)}$. If $a < 1$, the chain moves to $\tilde{\tau}_{(t)}$ with probability a; otherwise, it stays at $\tau_{(t)}$.

Once the sampling is complete for each chain, we sample permutations from the obtained set of per-topic permutations $(\tau_{(1)}, \tau_{(2)}, \ldots, \tau_{(T)})$ with probability $\propto \alpha'_t \exp\left(\sum_{i=1}^{K} \theta'_{t,i}\, d(\tau_{(t)}, \sigma_i)\right)$. The underlying assumption for this approximate sampling algorithm is that the probability mass of a mixture component is proportional to the probability of a sample generated from this component. The sampling procedure is repeated for all Q examples, and the average per-type distance is used as the estimate of the right-hand side of (9).

This procedure is easily extended to top-k lists: when forming the next chain element $\tilde{\tau}_{(t)}$, we may either transpose two elements of $\tau_{(t)}$, or replace one of its elements with an element not in $\tau_{(t)}$ (i.e. from the pool of all elements in the constituent rankings of the given example).

While convergence results have been presented in statistics literature (e.g. [Diaconis and Saloff-Coste, 1998]) for some distances when sampling from (3), no results are known for the extended models (4) and (7) we have considered in this work. However, we found experimentally that chains converge rapidly for the two settings we are considering. Moreover, as the chain proceeds, we only need to compute the incremental change in distance due to a single swap at each step, which results in substantial computational savings.

Estimating $\boldsymbol{\alpha}$

Following (8), we estimate the type mixture coefficients $\boldsymbol{\alpha}$ as the proportions of the types in all of the Q sampled rankings (with the additional pseudocounts β).

Solving the left-hand side of (9) **for $\boldsymbol{\theta}$**

As we noted in Section 2.2, solving $E_\theta(D) = \overline{D}$ under (3) for θ may be efficient for *decomposable* distance functions. Indeed, for (decomposable) Kendall's tau distance over permutations, $E_\theta(D_K)$ is efficient to compute and is monotone decreasing, so line search for θ converges quickly [Fligner and Verducci, 1986]. An analogous result was derived by [Klementiev *et al.*, 2008] for the augmented Kendall's tau over top-k lists (Section 2.1). Thus, requiring distance functions to satisfy the decomposability property may enable us to solve the left-hand side of (9) efficiently.

4.2 Inference

We use the sampling procedure described in Section 4.1 during inference to estimate the most likely permutation for a given set of votes.

5 Evaluation

In the absence of available labeled ranked data exhibiting domain variability, we construct our own data sets representative of a realistic application scenario. We evaluate the proposed framework on two types of rankings we have considered so far: permutations and top-k lists.

5.1 Aggregating Typed Permutations

We first consider the case of rank aggregation for typed permutations using Kendall's tau as the distance function in (7).

For this first set of experiments, we considered $K = 10$ judges, producing rankings over $n = 30$ objects for $Q = 100$ examples. Each example was associated with one of $T = 5$ types, according to $\boldsymbol{\alpha}^* = (0.4, 0.2, 0.2, 0.1, 0.1)$. Roughly half of the judges are chosen to be experts (i.e. produce good rankings) for each of the T types.

More precisely, the votes of individual judges were produced by sampling models (3), with the same location parameter $\sigma^* = e$ (an identity permutation over $n = 30$ objects). We chose their concentration parameters as follows: we first flip a coin to decide whether or not the i-th ranker is an expert for type t. If the ranker is an expert, its parameter $\theta^*_{t,i}$ is randomly chosen from a small interval close to -1, otherwise it is chosen to be around -0.05.

At the end of each EM iteration, we sampled the current model (7), and computed the Kendall's tau distance between the generated permutation π and the true permutation σ^* for each of the Q examples, and report the performance in terms of the average distance.

In addition to the sampling procedure we proposed to estimate the right-hand side of (9), we also tried using the true permutation σ^* along with the corresponding true type t^* in place of the sampled values to see how well the learning procedure can do.

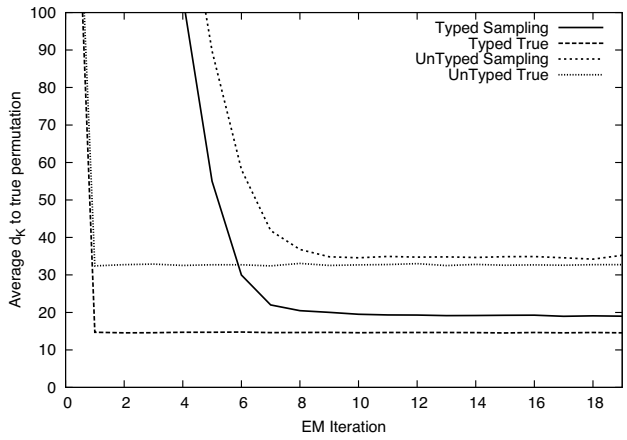

Figure 1: Learning performance over permutations when RHS is estimated using sampling (Sampling), or the true permutation (True). All results are averaged over 10 runs. Our model (Typed) significantly outperforms the type agnostic model (4) (UnTyped).

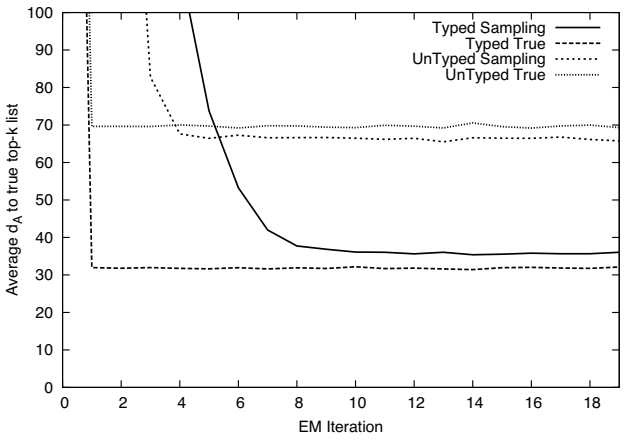

Figure 2: Learning performance over top-k lists when RHS is estimated using sampling (Sampling), or the true top-k list (True). All results are averaged over 10 runs. Our model (Typed) significantly outperforms the type agnostic model (4) (UnTyped).

Figure 1 shows that our model significantly outperforms the type agnostic model proposed in [Klementiev *et al.*, 2008] for this setting. While the type agnostic model *(UnTyped Sampling)* achieves an average distance of 34, our model *(Typed Sampling)* requires an average distance of 19, representing about 44% reduction in the number of adjacent transpositions at convergence. Moreover, the model converges quickly, and its performance approaches the case when true permutations and their corresponding types are known.

5.2 Aggregating Typed Top-K Lists

We now consider the case of combining typed top-k lists using the augmented Kendall's tau distance (see Section 2.1) in (7). The setup and the data was produced similarly to permutations experiments in Section 2.1. However, when generating top-k, $k = 30$ objects were selected from a pool of $n = 100$.

We compare our top-k list instantiated model against the corresponding type agnostic model, reporting results in Figure 2 in terms of the average augmented Kendall's tau distance from the true top-k lists. Again, our framework significantly outperforms the type agnostic model, reducing the average augmented Kendall's tau distance to true lists by approximately 31, resulting in 47% reduction in the number of adjacent transpositions.

5.3 Discussion

In many practical applications, constituent rankers are likely to specialize in some input domains, while performing poorly in others. Both experiments demonstrate that the model we proposed can take advantage of the latent type information to learn to aggregate the votes of such rankers effectively. It produces better joint rankings than the type agnostic model of [Klementiev *et al.*, 2008].

The EM-based learning algorithm we propose is robust, converging quickly with just $Q = 100$ examples across multiple runs. It is worth emphasizing that the additional expressivity of the model we propose does not prevent it from learning from a small number of examples successfully. Additionally, the computational requirements scale linearly with the number of topics.

6 Relevant Work

Modeling ranked data is an extensively studied problem in statistics, information retrieval, and machine learning literature. Distance-based models with Kendall's and Spearman's metrics for fully ranked data were introduced and investigated in [Mallows, 1957]. A number of new metrics for partial rankings were since introduced and analyzed (e.g. [Critchlow, 1985; Estivill-Castro *et al.*, 1993; Fagin *et al.*, 2003]), and various extensions to the model itself have been proposed (e.g. [Fligner and Verducci, 1986]); see [Marden, 1995] for an excellent overview. [Murphy and Martin, 2003] used mixtures of Mallows models (3) to analyze ranking data from heterogeneous populations, and [Busse *et al.*, 2007] propose a method for clustering such data. A large body of work also exists on mixture models (or LCA, [Lazarsfeld and Henry, 1968]).

While a number of heuristic [Shaw and Fox, 1994; Dwork *et al.*, 2001] and supervised learning approaches [Liu *et al.*, 2007] exist for rank aggregation, few *learn* to combine rankings without supervision.

Most directly related to our work is the generalization to multiple input rankings proposed and studied in [Lebanon and Lafferty, 2002]; [Klementiev *et al.*, 2008] derived an EM-based algorithm to estimate its parameters. We extend their model to include the notion of domain expertise.

7 Conclusions

In this work, we propose an *unsupervised learning* framework for rank aggregation over votes of rankers with domain-specific expertise. We introduce a model, derive an EM-based algorithm to estimate its parameters, and propose methods to make learning efficient. Finally, we evaluate the framework on combining full rankings and on combining top-k lists, and demonstrate that it significantly and robustly outperforms the domain agnostic model proposed in [Klementiev *et al.*, 2008]. This approach is potentially applicable to many problems in Information Retrieval and Natural Language Processing, e.g. meta-search or aggregation of machine translation systems' output, where domain variability presents a major challenge [Koehn and Schroeder, 2007; Geng *et al.*, 2008]. Developing unsupervised techniques is particularly important as annotated data is very difficult to obtain for ranking problems, especially for multiple domains.

Acknowledgments

We thank Ming-Wei Chang, 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, Swiss NSF scholarship PBGE22-119276, and by MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC.

References

[Busse *et al.*, 2007] Ludwig M. Busse, Peter Orbanz, and Joachim M. Buhmann. Cluster analysis of heterogeneous rank data. In *Proc. of the International Conference on Machine Learning (ICML)*, 2007.

[Critchlow, 1985] Douglas E. Critchlow. *Metric Methods for Analyzing Partially Ranked Data*, volume 34 of *Lecture Notes in Statistics*. Springer-Verlag, 1985.

[Dempster *et al.*, 1977] 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*, 39:1–38, 1977.

[Diaconis and Graham, 1977] Persi Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society*, 39:262–268, 1977.

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

[Dwork *et al.*, 2001] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. of the International World Wide Web Conference (WWW)*, pages 613–622, 2001.

[Estivill-Castro *et al.*, 1993] Vladimir Estivill-Castro, Heikki Mannila, and Derick Wood. Right invariant metrics and measures of presortedness. *Discrete Applied Mathematics*, 42:1–16, 1993.

[Fagin *et al.*, 2003] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17:134–160, 2003.

[Fligner and Verducci, 1986] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society*, 48:359–369, 1986.

[Geng *et al.*, 2008] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum. Query dependent ranking using k-nearest neighbor. In *SIGIR*, pages 115–122, 2008.

[Hastings, 1970] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

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

[Klementiev *et al.*, 2008] Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggregation with distance-based models. In *Proc. of the International Conference on Machine Learning (ICML)*, 2008.

[Koehn and Schroeder, 2007] Philipp Koehn and Josh Schroeder. Experiments in domain adaptation for statistical machine translation. In *ACL 2007, Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007.

[Lazarsfeld and Henry, 1968] Paul F. Lazarsfeld and Neil W. Henry. *Latent Structure Analysis*. Houghton Mifflin, 1968.

[Lebanon and Lafferty, 2002] Guy Lebanon and John Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *Proc. of the International Conference on Machine Learning (ICML)*, 2002.

[Liu *et al.*, 2007] Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. Supervised rank aggregation. In *Proc. of the International World Wide Web Conference (WWW)*, 2007.

[Mallows, 1957] C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[Marden, 1995] John I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, 1995.

[Murphy and Martin, 2003] Thomas Brendan Murphy and Donal Martin. Mixtures of distance-based models for ranking data. *Computational Statistics & Data Analysis*, 41:645–655, 2003.

[Neal, 1996] Radford M. Neal. Sampling from multimodal distribution using tempered transitions. *Statistics and Computing*, 6:353–366, 1996.

[Rosti *et al.*, 2007] Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. Combining outputs from multiple machine translation systems. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 228–235, 2007.

[Shaw and Fox, 1994] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In *Text REtrieval Conference (TREC)*, pages 243–252, 1994.