Unsupervised Rank Aggregation with Distance-Based Models

Alexandre Klementiev, Dan Roth, and Kevin Small
University of Illinois at Urbana-Champaign
Motivation

- Consider a panel of judges
  - Each (independently) generates preferences over items, i.e. (partial) rankings

- The need to meaningfully aggregate their rankings is a fundamental problem
  - Applications are plentiful in Information Retrieval and Natural Language Processing
Meta-search

- **Meta-search**: combine results of multiple search engines into a single ranking
  - Sets of ranked pages are different across rankers
  - *Supervision is difficult to get*, often collected indirectly (e.g. clickthrough data)
Multilingual Named Entity Discovery

- **Named Entity Discovery [Klementiev & Roth, ACL 06]:** given a bilingual corpus one side of which is annotated with Named Entities, find their counterparts in the other.

### Candidate Rankings

<table>
<thead>
<tr>
<th>Candidate</th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>гуимареша</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>муаммар</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>гимараешу</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>футбол</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>гамма</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

- NEs are often transliterated: rank according to a transliteration model.
- NEs tend to co-occur across languages: rank according to temporal similarity.
- NEs tend to co-occur in similar contexts: rank according to contextual similarity.
- NEs tend to co-occur in similar topics: rank according to topic similarity.
- etc.
# Outline

- **Motivation**
- **Problem Statement**
- **Overview of our approach**
- **Background**
  - Mallows models
  - Extended Mallows models
- **Unsupervised Learning and Inference**
- **Instantiations of the framework**
  - Combining permutations
  - Combining top-k lists
- **Experiments**
- **Conclusions and Current/Future work**

**Introduction and background**

**Our contribution**
Problem

How can we combine (partial) object preferences from multiple judges into a joint ranking?

- In IR, many approaches (data fusion) aggregate rankings heuristically
  - Linear score/rank aggregation is frequently used
  - Assume domain knowledge is available
- Supervised machine learning techniques require labeled training data

For ranking problems, supervision is difficult to obtain
Overview of Our Approach

- We propose a formal framework for *unsupervised* rank aggregation
  - Judges independently generate a (partial) ranking attempting to reproduce the true underlying ranking based on their level of expertise
  - We derive an EM-based algorithm treating the votes of individual judges and the true ranking as the observed and unobserved data, respectively
  - We instantiate the framework for the cases of combining permutations and combining *top-k* lists
Concepts and Notation

- Permutation $\pi$ over $n$ objects $x_1 \ldots x_n$
  - $\pi(i)$ is the rank assigned to object $x_i$
  - $\pi^{-1}(j)$ is the index of the object assigned to rank $j$
  - $e = \pi\pi^{-1} = \pi^{-1}\pi$ is the identity permutation

- Set $S_n$ of all $n!$ permutations

- Distance $d : S_n \times S_n \rightarrow \mathbb{R}_+$ between permutations
  - E.g. Kendall’s tau distance: minimum number of adjacent transpositions needed to turn $\pi$ into $\sigma$

- $d$ is assumed to satisfy the right invariance property: does not depend on arbitrary re-labeling of the $n$ objects
  - $d(\pi, \sigma) = d(\pi\pi^{-1}, \sigma\pi^{-1}) = d(e, \nu) = D(\nu)$. If $\nu$ is a r.v., so is $D=D(\nu)$
Background: Mallows Models

- \( \theta \in \mathbb{R}, \theta \leq 0 \) is the dispersion parameter
- \( \sigma \in S_n \) location parameter
- \( d(\cdot, \cdot) \) right-invariant, \( sZ(\theta, \sigma) \) not depend on \( \sigma \)
- If \( D \) can be decompose \( D(\pi) = \sum_{i=1}^{m} V_i(\pi) \) where \( V_i \) are indep. r.v. \( \sim E_\theta(D) \) may be efficient to compute [Fligner and Verducci ‘86]
Generative Story for Aggregation

Generate the true $\pi$ according to prior $p(\pi)$

Draw $\sigma_1 \ldots \sigma_K$ independently from $K$ Mallows models $p(\sigma_i | \theta_i, \pi)$, with the same location parameter $\pi$

$$p(\pi, \sigma | \theta) = p(\pi) \prod_{i=1}^{K} p(\sigma_i | \theta_i, \pi)$$
Background: Extended Mallows Models

The associated conditional model (when votes of $K$ judges $\sigma \in S_n^K$ are available) proposed in [Lebanon and Lafferty '02]:

Free parameters $\theta \in \mathbb{R}^K$, $\theta \leq 0$ represent the degree of expertise of individual judges.

It is straightforward to generalize both models to partial rankings by constructing *appropriate distance functions*.
# Outline

- **Motivation**
- **Problem Statement**
- **Overview of our approach**
- **Background**
  - Mallows models
  - Extended Mallows models
- **Unsupervised Learning and Inference**
  - Instantiations of the framework
    - Combining permutations
    - Combining top-k lists
- **Experiments**
- **Conclusions and Current/Future work**

**Introduction and background**

**Our contribution**
Our Approach

- We propose a formal framework for unsupervised rank aggregation based on the extended Mallows model formalism.
- We derive an EM-based algorithm to estimate model parameters $\theta$.

- Observed data: votes of individual judges
- Unobserved data: true ranking
Learning

Denoting $\theta^t_0$ be the value of parameters from the previous iteration, the M step for the $i^{th}$ ranker is:

$\text{LHS} > (n!)$

In general, $> n!$ computations.

Average distance between votes $\pi_{1..Q}$ of the $i^{th}$ ranker and $\pi_{1..Q}$ of the unobserved data.
Learning and Inference

**Learning (estimating \( \theta \))**
- For \( K \) constituent rankers, repeat:
  - Estimate the RHS given current parameter values \( \theta \)
    - Sample with Metropolis-Hastings
    - Or use heuristics
  - Solve the LHS to update \( \theta \)
    - Efficient estimation can be done for particular types of distance functions

**Inference (computing the most likely ranking)**
- Sample with Metropolis-Hastings or use heuristics as above

*Depends on ranking type, more about this later*
We have *not committed* to a particular type of ranking.

In order to instantiate the framework:

- Design a distance function appropriate for the setting
  - If a function if right invariant and decomposable \([LHS]\) estimation can be done quickly (more about this later)
- Design a sampling procedure for learning \([RHS]\) and inference
Case 1: Combining Permutations [LHS]

- Kendall tau distance $D_K$ is the minimum number of adjacent transpositions needed to transform one permutation into another.

- Can be decomposed into a sum of independent random variables:

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

where

$$V_i(\pi) = \sum_{j>i} I(\pi^{-1}(i) - \pi^{-1}(j))$$

- And the expected value can be shown to be:

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

Monotonically decreasing, can find $\theta$ with line search.
Case 1: Combining Permutations

Sampling from the base chain of random transpositions

- Start with a random permutation $\pi$
- If $\alpha = p(\pi' | \theta, \sigma) / p(\pi | \theta, \sigma) \geq 1$, i.e., two objects forming
  - If $\pi'$, chain moves to $\pi'$
  - Else, chain moves to $\pi'$ with probability

- Note that we can compute distance incrementally, i.e. add the change due to a single transposition

- Convergence
  - $n \log(n)$ if $d$ is Cayley’s distance [Diaconis ’98], likely similar for some others
  - No convergence results for general case, but it works well in practice
Case 1: Combining Permutations [RHS]

An alternative heuristic: weighted Borda count, i.e.

- Linearly combine ranks of each object and argsort
- Model parameters $\theta$ represent relative expertise, so it makes sense to weigh rank $e^{(-\theta_i)}_i =$

$$w_1 + w_2 + \ldots + w_K$$
Case 2: Combining Top-k [LHS]

- We extend Kendall tau to top-k

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

- Bring grey boxes to bottom
- Switch with objects in (k+1)
- Kendall’s tau for the k elements
Case 2: Combining Top-k \([\text{LHS} \& \text{RHS}]\)

- R.v.’s \(\tilde{V}_i\) and \(\tilde{U}_i\) are independent, we can use the same trick to show that [LHS] is:

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

- Also monotonically decreasing, can again use line search
- Both \(\tilde{D}_K\) and \(E_\theta(\tilde{D}_K)\) reduce to Kendall tau results when same elements are ranked in both lists, i.e. \(r = 0\)

- Sampling / heuristics for [RHS] and inference are similar to the permutation case
Outline

- Motivation
- Problem Statement
- Overview of our approach
- Background
  - Mallows models
  - Extended Mallows models
- Unsupervised Learning and Inference
  - Instantiations of the framework
    - Combining permutations
    - Combining top-k lists
- Experiments
- Conclusions and Current/Future work
Exp. 1 Combining permutations

- **Judges**: $K = 10$ (Mallows models)
- **Objects**: $n = 30$
- **Q**: 10 sets of votes

Using sampling to estimate the RHS

Using weighted Borda heuristic to estimate the RHS

Using true rankings to evaluate the RHS
Exp. 2 Meta-search dispersion parameters

- **Judges:** $K = 4$ search engines (S1, S2, S3, S4)
- **Documents:** Top $k = 100$
- **Queries:** $Q = 50$ queries

Define Mean Reciprocal Page Rank (MRPR): mean rank of the page containing the correct document

- Our model gets 0.92

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>-0.065</td>
<td>0.0</td>
<td>-0.066</td>
<td>-0.049</td>
</tr>
<tr>
<td>MRPR</td>
<td>0.86</td>
<td>0.43</td>
<td>0.82</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Model parameters correspond to ranker quality
Exp. 3 Top-k rankings: robustness to noise

- **Judges:** $K = 38$ TREC-3 ad-hoc retrieval shared task participants
- **Documents:** Top $k = 100$ documents
- **Queries:** $Q = 50$ queries

Replaced $K_r \in [0, K]$ randomly chosen participants with random rankers. Baseline: rank objects according to score:

$$\text{CombMNZ}_{\text{rank}} = N_x \times \sum_{i=1}^{K} (k - r_i(x, q))$$

where $r_i(x, q)$ is the rank of $x$ returned by $i$ for query $q$, $\varepsilon N_x$ is the number of participants with $x$ in top-$k$
Exp. 3 Top-k rankings: robustness to noise

Learn to discard random rankers without supervision
Conclusions

- Propose a formal mathematical and algorithmic framework for aggregating (partial) rankings \textit{without} supervision
  - Show that learning can be made efficient for decomposable distance functions

- Instantiate the framework for combining \textit{permutations} and combining \textit{top-k} lists
  - Introduce a novel distance function for \textit{top-k}
Future work / work in progress

- Instantiate to other types of partial rankings
  - E.g. MT system aggregation: combine alignments

- Query / document type dependence: experts quality may depend on types of queries or objects being ranked

- Position dependence:
  - Right-invariant $d$, which is position-dependent (e.g. favors agreement at the top), need to be able to simplify the LHS.
  - [Flinger, Verducci ‘86] propose a multistage extension where objects are chosen going from top down position dependent $\theta$. Can we combine the ideas?

- Domain Adaptation