Unsupervised Rank Aggregation with Distance-Based Models

Collaborators:  
Alex Klementiev (Johns Hopkins University)  
Ivan Titov (Saarland University)  
Dan Roth (University of Illinois)  

Kevin Small  
Tufts University
Motivation

- Consider a panel of judges
  - Each independently generates (partial) rankings over objects to the best of their ability

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

<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>27</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>гимараешу</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>футбол</td>
<td>59</td>
<td>2</td>
<td>31</td>
<td>12</td>
</tr>
<tr>
<td>гамма</td>
<td>7</td>
<td>17</td>
<td>51</td>
<td>32</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- NEs are often transliterated: rank according to a *transliteration model score*.
- NEs tend to co-occur across languages: rank according to *temporal alignment*.
- 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.
Overview of Our Approach

- We propose a formal framework for *unsupervised* structured label aggregation
  - Judges independently generate a (partial) labeling attempting to reproduce the true underlying label based on their expertise in a given domain
  - We derive an EM-based algorithm treating the votes of individual judges and the true label as the observed and unobserved data, respectively

- *Intuition*: experts in a given domain are better at generating votes close to true ranking and will tend to agree with each other, while the non-experts will not

- We instantiate the framework for the cases of combining *permutations*, combining *top-k lists*, and combining *dependency parses*. 
Notation

- Permutation \( \pi \) over \( n \) objects \( x_1 \ldots x_n \)
  - \( e = (1,2,\ldots,n) \) is the identity permutation
- Set \( S_n \) of all \( n! \) permutations
- Distance \( d : S_n \times S_n \to \mathbb{R}_+ \) between permutations
  - E.g. Kendall’s tau distance: minimum number of adjacent transpositions
    \[
    d_K\left(\begin{array}{c}2 \\ 1 \\ 3 \\ 4 \end{array}, \begin{array}{c}1 \\ 2 \\ 4 \\ 3 \end{array}\right) = 3
    \]
- \( d \) is assumed to be invariant to arbitrary re-labeling of the \( n \) objects
  - \( d(\pi, \pi') = d(e, \pi) = D(\pi) \). If \( \pi \) is a r.v., so is \( D=D(\pi) \)
    \[
    d_K\left(\begin{array}{c}2 \\ 1 \\ 3 \\ 4 \\ 1 \\ 2 \\ 4 \end{array}, \begin{array}{c}1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 4 \end{array}\right) = d_K\left(\begin{array}{c}1 \\ 2 \\ 4 \\ 3 \\ 2 \\ 1 \\ 4 \end{array}, \begin{array}{c}2 \\ 1 \\ 3 \\ 4 \\ 4 \\ 2 \\ 1 \end{array}\right) = D_K(\begin{array}{c}2 \\ 1 \\ 3 \\ 4 \end{array}) = 3
    \]
Background: Mallows Models

\[
p(\pi|\theta, \sigma) = \frac{1}{Z(\theta, \sigma)} \exp(\theta \cdot d(\pi, \sigma))
\]

where 
\[
Z(\theta, \sigma) = \sum_{\pi \in S_n} \exp(\theta \cdot d(\pi, \sigma))
\]

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

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

Draw $\xi_1, \ldots, \xi_K$ independently from $K$ Mallows models $p(\xi_i | \pi, \sigma_i, \theta)$, with the same location parameter $\xi$

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

The associated conditional model (when votes of $K$ judges $\sigma \in \mathcal{S}_n^K$ are available) proposed in [Lebanon and Lafferty ’02]:

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

where $Z(\theta, \sigma) = \sum_{\pi \in \mathcal{S}_n} p(\pi) \exp\left( \sum_{i=1}^{K} \theta_i d(\pi, \sigma_i) \right)$

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
  - Incorporating domain-specific expertise
  - Instantiations of the framework
    - Combining permutations / top-k lists

- Experiments
- Dependency Parsing
- Conclusions
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 $\theta'$ to be the value of parameters from the previous iteration, the M step for the $i^{th}$ ranker is:

\[
E_{\theta_i}(D) = \sum_{(\pi^{(1)}, \ldots, \pi^{(Q)}) \in S_n^Q} \left( \frac{1}{Q} \sum_{q=1}^{Q} d(\pi^{(q)}, \sigma_i^{(q)}) \right) \prod_{j=1}^{Q} p(\pi^{(j)} | \theta', \sigma^{(j)})
\]

In general, $> n!$ computations

Average distance between votes of the $i^{th}$ ranker and $\pi^{(1..Q)}$

Marginal of the unobserved data $\pi^{(1..Q)}$
Learning and Inference

For $K$ constituent rankers, repeat:

- Estimate the RHS given current parameter values.
  - Sample with Metropolis-Hastings
  - Or use heuristics
- Solve the LHS to update.
  - Efficient estimation can be done for particular types of distance functions

Inference (computing the most likely ranking)
- Sample with Metropolis-Hastings or use heuristics
Domain-specific expertise?

- Relative expertise may **not stay the same**
  - May depend on the type of objects
  - May depend on the type of query

- Typically, ranked supervised data to estimate judges’ expertise is **very expensive to obtain**
  - Especially for multiple types
Mallows Models with Domain-Specific Expertise

The associated conditional model (when votes of $K$ judges $\sigma \in S_n^K$ are available) can be derived:

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

Free parameters $\theta \in \mathbb{R}^{T \times K}, \theta \leq 0$ represent the degree of expertise of individual judges. $\alpha \in \mathbb{R}^{T}$ are the mixture weights.

Note: it is straightforward to generalize these models to other structured labels (e.g. partial rankings) by constructing appropriate distance functions.
For each of $i^{th}$ ranker and $t^{th}$ type:

- Estimate $\alpha_t$ (1) and $E_{\theta_t,i}(D)$ (2) given current parameter values $\theta'$ and $\alpha'$
- Solve 3 to update $\theta_{t,i}$
- Repeat
Instantiating the Framework

- We have *not committed* to a particular type of structure
- In order to instantiate the framework:
  - Design a distance function appropriate for the setting
    - If a function is right invariant and decomposable [LHS] estimation can be done quickly
  - 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:

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

- 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 quickly.
Case 1: Combining Permutations [RHS]

Sampling from the base chain of random transpositions

- Start with a random permutation

- If chain is at $\pi$, randomly transpose two objects forming $\pi'$
  - If $a = p(\pi' | \theta, \sigma) / p(\pi | \theta, \sigma) \geq 1$ chain moves to $\pi'$
  - Else, chain moves to $\pi'$ with probability $a$

- 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
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 rankers as $w_i = e^{-\theta_i}$

\[
\begin{align*}
 w_1 + w_2 + \ldots + w_K \xrightarrow{\text{argsort}}
\end{align*}
\]
Case 2: Combining Top-k [LHS]

- We extend Kendall tau to top-k

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

- Bring grey boxes to bottom
- Switch with objects in \((k+1)\)
- Kendall’s tau for the \(k\) elements

\[r\) grey boxes \]
\[z\) white boxes \]
\[r + z = k\]
Case 2: Combining Top-k [LHS & 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
  - Incorporating domain-specific expertise
- Instantiations of the framework
  - Combining permutations / top-k lists
- Experiments
- Dependency Parsing
- Conclusions
Exp. 1 Combining permutations

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

Using sampling to estimate the RHS

Using true rankings to evaluate the RHS

Using weighted $(e^{-\theta_i})$ Borda heuristic to estimate 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:

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

where $r_i(x, q)$ is the rank of $x$ returned by $i$ for query $q$, and $N_x$ 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

Precision

Number of random rankers $K_r$
Outline

- Motivation
  - Problem Statement
  - Overview of our approach
  - Background
    - Mallows models / Extended Mallows models
- Introduction and background
  - Unsupervised Learning and Inference
    - Incorporating domain-specific expertise
  - Instantiations of the framework
    - Combining permutations / top-k lists
  - Experiments
  - Dependency Parsing
  - Conclusions
Dependency Parses

Let $v(i)$ denote a pair of head offset and label, i.e. a link.

Labeled attachment score is link accuracy, trivially computed from Hamming distance:

$$d_H(v, y) = \sum_{i=1}^{n} [v(i) \neq y(i)]$$
Parameter estimation of the type-agnostic model can be done directly.

Let us assume there are exactly $|S|$ possibilities for each link, and that $j^{th}$ (of $Q$) sentences has $n^{(j)}$ words, $\sum_{j=1}^{Q} n^{(j)} = N$.

On each round of training the learning procedure for the type-agnostic model is equivalent to:

$$\theta_i = \log R_i' - \log(1 - R_i') - \log(|S| - 1)$$

$$R_i' = \frac{1}{N} \sum_{j=1}^{Q} \sum_{l=1}^{n^{(j)}} \frac{\sum_{v \in S} \mathbb{1}[v \neq y^{(j)}_{i,(l)}] \exp \left( \sum_{i=1}^{K} \theta_i' \mathbb{1}[v \neq y^{(j)}_{i,(l)}] \right)}{\sum_{v \in S} \exp \left( \sum_{i=1}^{K} \theta_i' \mathbb{1}[v \neq y^{(j)}_{i,(l)}] \right)}$$

With small $|S|$, parameter estimation can be done quickly!
Dependency Parses: Aggregation

- Dependency parsers are CoNLL-2007 shared task participants
  - 10 languages: Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian, and Turkish
  - 131 to 690 sentences and 4513 to 5390 words, depending on the language
  - Between 20 and 23 systems, depending on the language

- Varied the number of participants attempting to represent expertise in the entire pool

- Baseline: majority vote on each link (ties broken randomly)
Estimation of relative expertise correlates with true relative expertise

Results for Italian

Labeled Attachment

<table>
<thead>
<tr>
<th>Participant</th>
<th>$\theta$</th>
<th>True Rank</th>
<th>True Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jni@msi.vxu.se">jni@msi.vxu.se</a></td>
<td>-2.54</td>
<td>1</td>
<td>84.40</td>
</tr>
<tr>
<td><a href="mailto:sagae@is.s.u-tokyo.ac.jp">sagae@is.s.u-tokyo.ac.jp</a></td>
<td>-2.52</td>
<td>2</td>
<td>83.91</td>
</tr>
<tr>
<td><a href="mailto:nakagawa378@oki.com">nakagawa378@oki.com</a></td>
<td>-2.29</td>
<td>3</td>
<td>83.61</td>
</tr>
<tr>
<td><a href="mailto:johan.hall@vxu.se">johan.hall@vxu.se</a></td>
<td>-2.28</td>
<td>5</td>
<td>82.48</td>
</tr>
<tr>
<td><a href="mailto:carreras@csail.mit.edu">carreras@csail.mit.edu</a></td>
<td>-2.21</td>
<td>4</td>
<td>83.46</td>
</tr>
<tr>
<td><a href="mailto:chenwl@nict.go.jp">chenwl@nict.go.jp</a></td>
<td>-2.13</td>
<td>7</td>
<td>82.04</td>
</tr>
<tr>
<td><a href="mailto:attardi@di.unipi.it">attardi@di.unipi.it</a></td>
<td>-2.01</td>
<td>8</td>
<td>81.34</td>
</tr>
<tr>
<td><a href="mailto:xyduan@nlpira.ac.cn">xyduan@nlpira.ac.cn</a></td>
<td>-2.00</td>
<td>9</td>
<td>80.75</td>
</tr>
<tr>
<td><a href="mailto:ivan.titov@cui.unige.ch">ivan.titov@cui.unige.ch</a></td>
<td>-1.95</td>
<td>6</td>
<td>82.26</td>
</tr>
<tr>
<td><a href="mailto:dasmith@jhu.edu">dasmith@jhu.edu</a></td>
<td>-1.92</td>
<td>10</td>
<td>80.69</td>
</tr>
<tr>
<td><a href="mailto:michael.schiehlen@ims.uni-stuttgart.de">michael.schiehlen@ims.uni-stuttgart.de</a></td>
<td>-1.83</td>
<td>11</td>
<td>80.46</td>
</tr>
<tr>
<td><a href="mailto:bcbbb@db.csie.ncu.edu.tw">bcbbb@db.csie.ncu.edu.tw</a></td>
<td>-1.79</td>
<td>12</td>
<td>78.79</td>
</tr>
<tr>
<td><a href="mailto:prashanth@research.iit.ac.in">prashanth@research.iit.ac.in</a></td>
<td>-1.76</td>
<td>13</td>
<td>78.67</td>
</tr>
<tr>
<td><a href="mailto:richard@cs.lth.se">richard@cs.lth.se</a></td>
<td>-1.67</td>
<td>14</td>
<td>77.55</td>
</tr>
<tr>
<td><a href="mailto:nguyenml@jaist.ac.jp">nguyenml@jaist.ac.jp</a></td>
<td>-1.51</td>
<td>16</td>
<td>75.06</td>
</tr>
<tr>
<td><a href="mailto:joyce840205@gmail.com">joyce840205@gmail.com</a></td>
<td>-1.50</td>
<td>17</td>
<td>74.65</td>
</tr>
<tr>
<td><a href="mailto:s.v.m.canisius@uvt.nl">s.v.m.canisius@uvt.nl</a></td>
<td>-1.49</td>
<td>15</td>
<td>75.57</td>
</tr>
<tr>
<td><a href="mailto:francis.maes@lip6.fr">francis.maes@lip6.fr</a></td>
<td>-1.30</td>
<td>18</td>
<td>73.63</td>
</tr>
<tr>
<td><a href="mailto:zeman@ufal.mff.cuni.cz">zeman@ufal.mff.cuni.cz</a></td>
<td>-0.68</td>
<td>19</td>
<td>62.13</td>
</tr>
<tr>
<td><a href="mailto:svetoslav.marinov@his.se">svetoslav.marinov@his.se</a></td>
<td>-0.49</td>
<td>20</td>
<td>59.75</td>
</tr>
</tbody>
</table>
Dependency Parses: Aggregation

- Performance measured as average accuracy over 10 languages

- Largest improvement when fewer experts (higher practical significance)

- Number of good experts grows: voted baseline is harder to beat
Conclusions

- Propose a formal mathematical and algorithmic framework for aggregating (partial) structured labels without supervision
  - Show that learning can be made efficient for decomposable distance functions

- Instantiate the framework for combining permutations, combining top-k lists, and dependency parses
  - Introduce a novel distance function for top-k and dependency parses
Thanks!