Unsupervised Rank Aggregation with Distance-Based Models

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

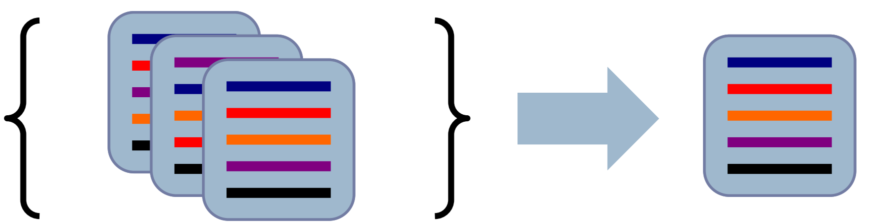
icml2008@helsinki.fi

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. clickthough 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

guimaraes

Candidate	r1	r2	r3	r4
гуимареша	1	1	3	3
муаммар	3	27	4	14
гимараешу	2	3	1	1
футбол	59	2	31	12
гамма	7	17	51	32
…				

  - NEs are often transliterated: ran … *ion 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.

Outline

- Motivation
- Problem Statement
- Overview of our approach
- Background
  - Mallows models
  - Extended Mallows models

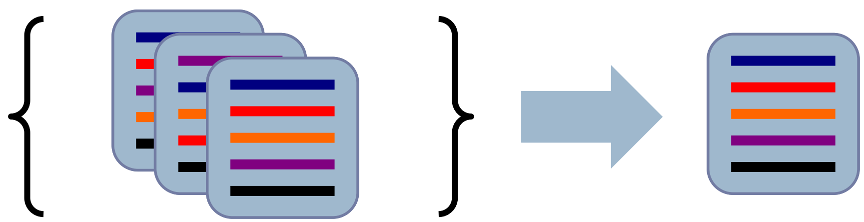
Introduction and background

- Unsupervised Learning and Inference
- Instantiations of the framework
  - Combining permutations
  - Combining top-k lists
- Experiments
- Conclusions and Current/Future work

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 π 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 $d : S_n \times S_n \to R_+$ between permutations
  - E.g. *Kendall's tau distance*: minimum number of adjacent transpositions needed to turn π into σ
- 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 ν is a r.v., so is $D=D(\nu)$

Background: Mallows Models

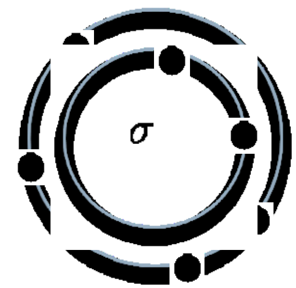
Uniform when $\theta = 0$

"Peaky" when $|\theta|$ is large

Expensive to compute

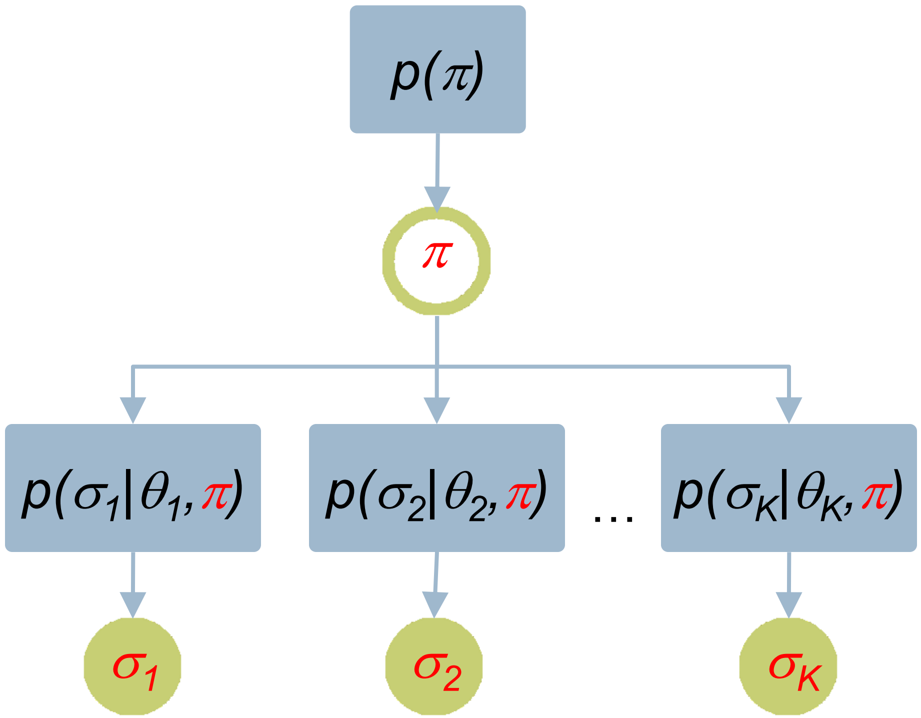
[redacted]

where [redacted]



- $\theta \in \mathbb{R},\ \theta \leq 0$ is the *dispersion* parameter
- $\sigma \in \mathcal{S}_n$ *location* parameter
- *d(·,·)* right-invariant, s $Z(\theta, \sigma)$ not depend on σ
- If *D* can be decompose $D(\pi) = \sum_{i=1}^{m} V_i(\pi)$ whe V_i are indep. r.v $E_\theta(D)$ en may be efficient to compute *[Fligner and Verducci '86]*

Generative Story for Aggregation



Generate the *true* π 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* π

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

Background: Extended Mallows Models

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

[redacted]

where [redacted]

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

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

Outline

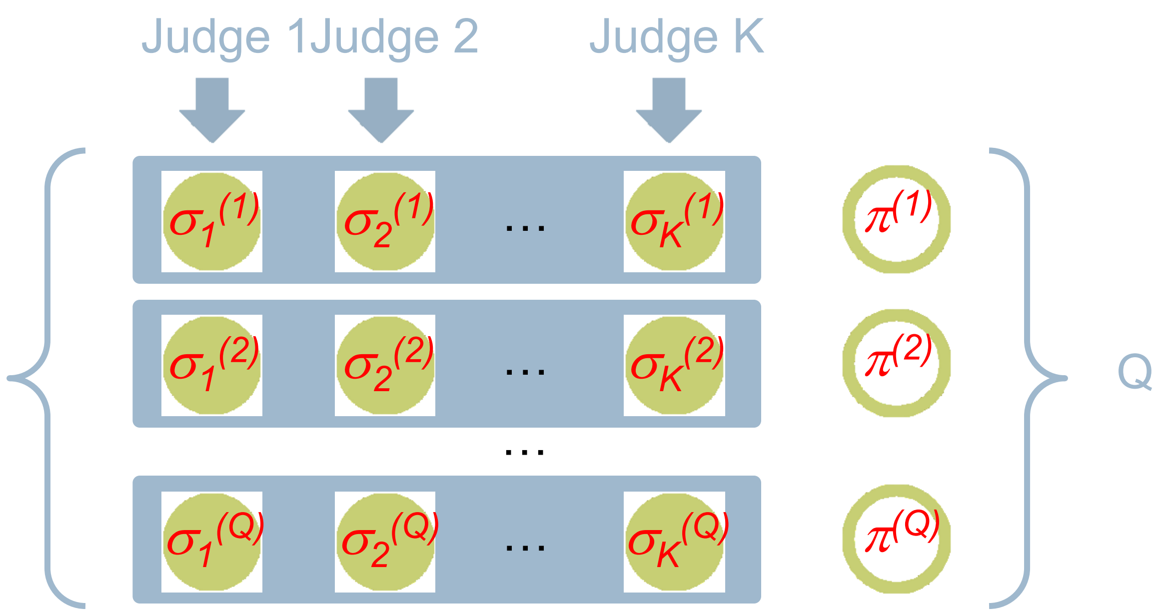
- Motivation
- Problem Statement
- Overview of our approach
- Background
  - Mallows models
  - Extended Mallows models
- Unsupervised Learning and Inference
- Instantiations of the framework
  - Combining permutations
  - Combining top-k lists
- Experiments
- Conclusions and Current/Future work

Introduction and background

Our contribution

Our Approach

- We propose a formal framework for <u>unsupervised rank aggregation</u> based on the *extended Mallows model* formalism
- We derive an EM-based algorithm to estimate model parameters $\boldsymbol{\theta}$
  - Observed data: votes of individual judges
  - Unobserved data: true ranking



Learning

Denoting θ' to be the value of parameters from the previous iteration, the M step for the i^{th} ranker is:

[redacted]

LHS

RHS

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

[redacted]

LHS **RHS**

<u>Learning (estimating θ)</u>

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

Depends on ranking type, more about this later

<u>Inference (computing the most likely ranking)</u>

- Sample with Metropolis-Hastings or use heuristics as above

Instantiating the Framework

- 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

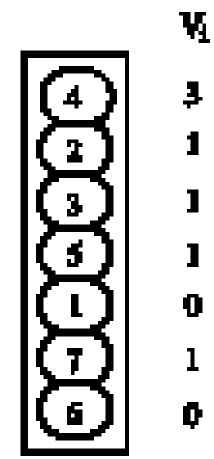
Case 1: Combining Permutations [LHS]

- 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:

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

- And the expected value can be shown to be:



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

Monotonically decreasing, can find θ *with line search*

Case 1: Combining Permutations [RHS]

Sampling from the base chain of random transpositions

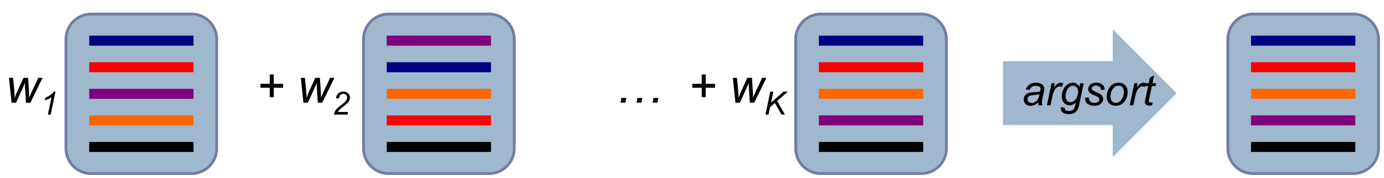
- Start with a random permutation π
- If ch $a = p(\pi'|\theta,\sigma)/p(\pi|\theta,\sigma) \geq 1$ se two objects π' orming π'
  - If π' a chain moves to
  - Else, chain moves to with probability
- Note that we can compute distance incrementally, i.e. add the change due to a single transposition
- Convergence
  - *nlog(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

COGNITIVE COMPUTATION GROUP
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Case 1: Combining Permutations [RHS]

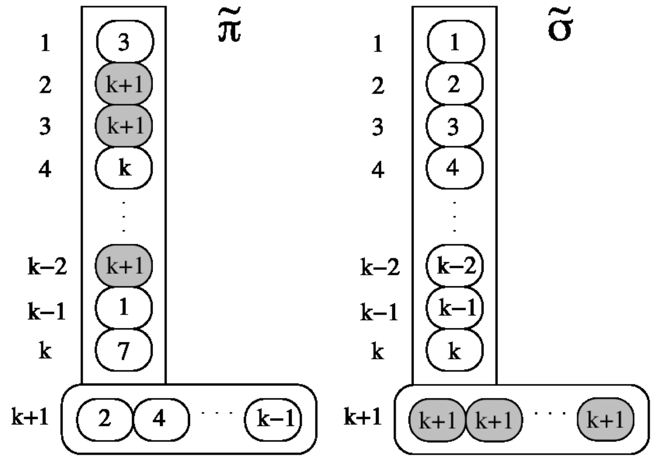
An alternative heuristic: weighted *Borda*count, i.e.

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



Case 2: Combining Top-k [LHS]

- We extend Kendall tau to top-k



r grey boxes
z white boxes
r + *z* = *k*

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

Bring grey boxes to bottom

Switch with objects in (k+1)

Kendall’s tau for the k elements

Case 2: Combining Top-k [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
- Instantiations of the framework
  - Combining permutations
  - Combining top-k lists
- Experiments
- Conclusions and Current/Future work

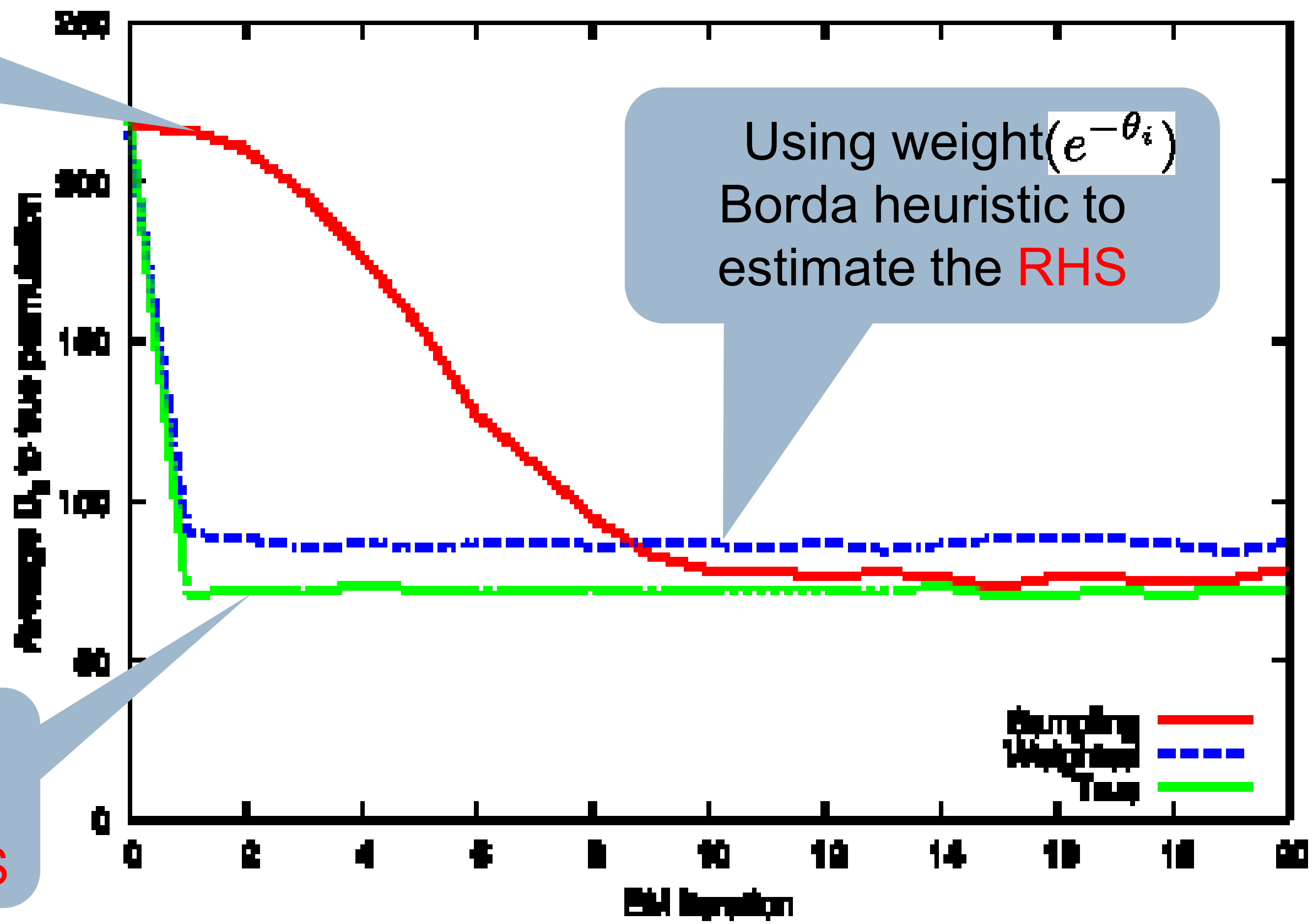
Introduction and background

Our contribution

Exp. 1 Combining permutations

Using sampling to estimate the RHS

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



Using weight $(e^{-\theta_i})$ 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***

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

Model parameters correspond to ranker quality

Exp. 3 Top-k rankings: robustness to noise

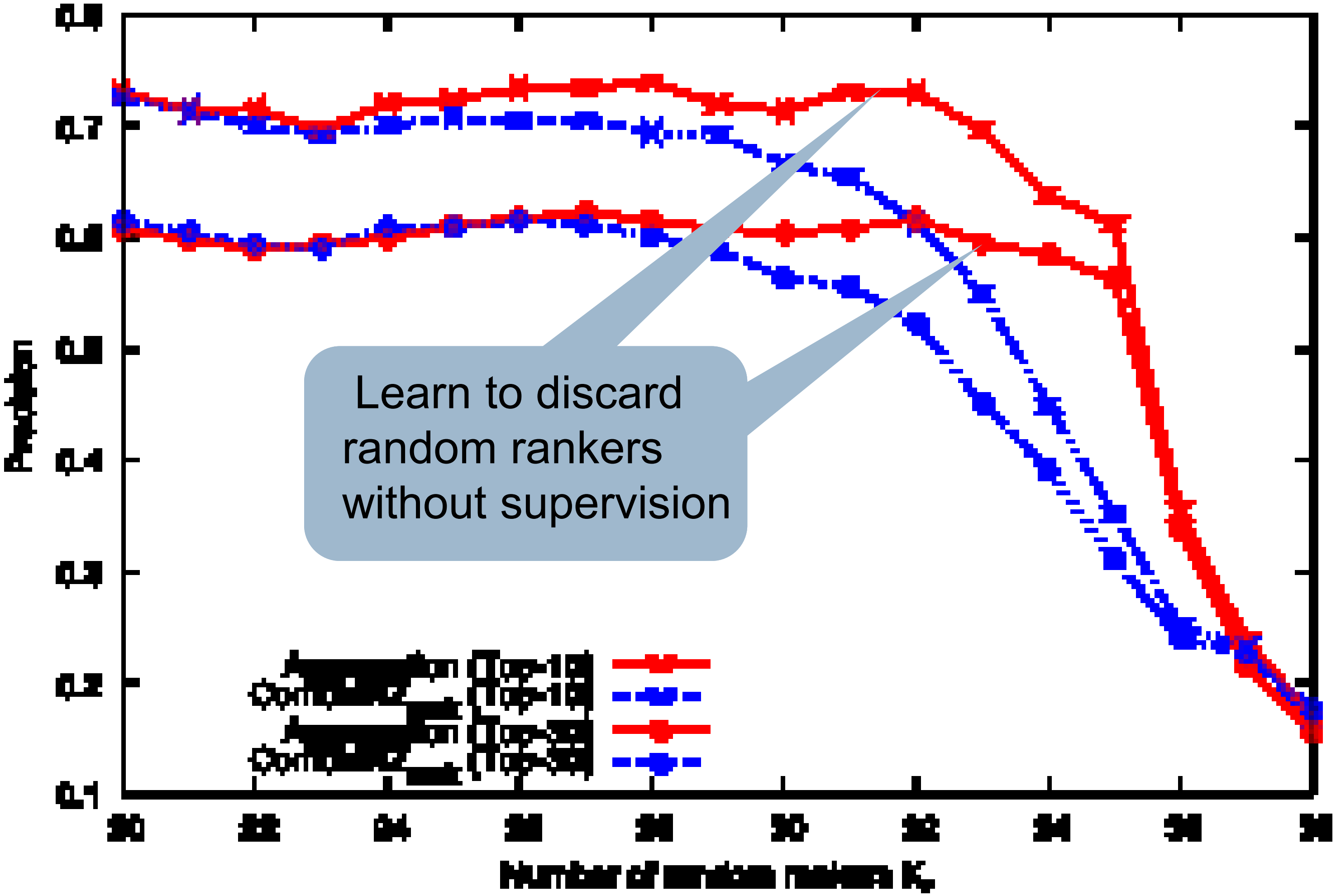
- *Judges: K* = 38 TREC-3 ad-hoc retrieval shared task participants
- *Documents*: Top *k* = 100 documents
- *Queries: Q* = 50 queries

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



Conclusions

- Propose a formal mathematical and algorithmic framework for aggregating (partial) rankings *without* supervision
  - Show that learning can be made efficient for decomposable distance functions
- Instantiate the framework for combining *permutations* and combining *top-k*lists
  - Introduce a novel distance function for *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 θ. *Can we combine the ideas?*
- Domain Adaptation