Minimising the Rank Aggregation Error

Mathijs M. de Weerdt
Delft University of Technology, Netherlands
m.m.deweerdt@tudelft.nl

Enrico H. Gerding
University of Southampton, UK
eg@ecs.soton.ac.uk

Sebastian Stein
University of Southampton, UK
ss2@ecs.soton.ac.uk

ABSTRACT

Rank aggregation is the problem of generating an overall ranking from a set of individual votes. The aim in doing so is to produce a ranking which is as close as possible to the (unknown) correct ranking for a given distance measure such as the Kendall-tau distance. The challenge is that votes are often both noisy and incomplete. Existing work has largely focused on finding the most likely ranking for a particular noise model (such as Mallows'). Instead, here we focus on minimising the error, i.e., the expected distance between the aggregated ranking and the true underlying one. Specifically, we show that the two objectives result in different rankings, and that these differences become especially significant when many votes are missing. Furthermore, we show how to compute local improvements on existing rankings to reduce the expected error. Finally, we run extensive experiments on both synthetic and real data to compare different aggregation rules. In particular, a surprising result is that for votes generated according to the Mallows' model, Copeland often outperforms Kemeny optimal, despite the latter being the maximum likelihood estimator.

Categories and Subject Descriptors

I.2.11 [**Distributed AI**]: Multiagent systems

Keywords

Economic paradigms: Social Choice Theory

1. INTRODUCTION

Rank aggregation is the problem of producing a complete ranking from votes cast by individual agents, where the votes can be seen as noisy and incomplete estimates of a ranking that is an underlying ground truth. This perspective on voting dates back to Marquis de Condorcet [25], who said that voting may be regarded as a way of uncovering this ground truth. There are many practical examples[1] of rank aggregation, including websites that produce rankings of restaurants, books and movies based on crowdsourced contributions from their users, scientific communities that use votes from their members to select which project proposals to fund or which papers to accept [18, 5], or peer grading in massive online open courses [3]. Another prominent application is the use of rank aggregation to produce a meta search engine from the search results of individual search engines [13]. In these settings votes are not only noisy, but also incomplete since typically only a subset of the candidates (e.g., restaurants or websites) is ranked by any single individual.

To find a ranking which is close to the ground truth, most current work assumes a probabilistic noise model such as Mallows [22, 32], and then aims to maximise the likelihood of an aggregate ranking. In Mallows' model, a probability is assumed for ordering a pair of candidates correctly, and votes are produced by repeatedly ordering all pairs until this results in a consistent (acyclic) ranking. For this model, it has been shown that Kemeny's rule is the maximum likelihood estimator (MLE) [32]. Similarly, some other commonly used voting rules are MLEs for specific noise models [9, 8].

However, in most settings the aim should arguably not be to find a most likely explanation of the noisy observations, but to find a ranking that gives the best results when used in subsequent decision making. When votes are noisy and incomplete, many rankings may have a likelihood of similar magnitude, and there may even be multiple rankings with the maximum likelihood. In all these cases, the probability that a ranking with maximum likelihood is the true ranking is small. When an aggregated ranking is used, success does not depend on having found the true ranking exactly. Rather, it is important to construct and use a ranking for which the distance to the true ranking is as small as possible *in expectation*. This means that we should aim to minimise the expected distance (for a particular measure) of an aggregate ranking to the true ranking — which we term the *error* — instead of aiming for a ranking that maximises the likelihood. In contrast to MLEs, to date it is still unknown which commonly used voting rules perform best regarding this objective.

Against this background, in this paper we assume noise according to the Mallows' model, and for this model we make the following novel contributions. (i) We show that the MLE is not always minimising the error. (ii) We show that computing the error is #P-hard. (iii) We show that *local Kemenisation*, a computationally simple procedure for improving aggregated rankings in terms of their likelihood, also reduces the error, and (iv) through experiments on both synthetic and real data, we show how noise and incompleteness influence the performance of a large set of voting rules.

[1] `http://www.tripadvisor.com/Restaurants`, `http://www.goodreads.com/choiceawards`

This is an extended version of a two-page abstract accepted for the 15th International Conference on Autonomous Agents and Multiagent Systems.
Appears at: 3rd Workshop on Exploring Beyond the Worst Case in Computational Social Choice. May 9th-10th, 2016. Singapore.

The paper is structured as follows. In Section 2 we introduce notation, the distance measure used, and the model for noise and incompleteness, and we show that Kemeny's rule is the MLE also for the model including incompleteness. In Section 3 we then formally define the objective of minimising the error and show how this is different from maximising the likelihood. We give the hardness proof of computing the error and show that local Kemenisation reduces the error. In Section 4 we first introduce the voting rules and their adaptations to settings with incomplete rankings. Subsequently, we evaluate these rules under varying levels of noise and incompleteness for both synthetic votes as well as on two ranking data sets from PrefLib [24]. Section 5 discusses related work and Section 6 concludes.

2. MODEL

The aim in this paper is to find rank aggregation rules which minimise the expected error, where the error is given by the Kendall-tau distance to the true ranking (as defined below). Formally, let $A = \{1, 2, \ldots, m\}$ denote a set of candidates or *alternatives*, where $m = |A|$ is the number of alternatives to be ranked. In addition, let $N = \{1, 2, \ldots, n\}$ denote the set of n agents or voters. Each agent has an incomplete ranking over the set of candidates. This is modeled as a complete order on a subset of the candidates.[2] We thus define a vote by agent k as a ranking, i.e., linear order, over a subset $A_k \subseteq A$ of the candidates, denoted by $\sigma_k : A_k \to \{1, 2, \ldots, |A_k|\}$. Here, $\sigma_k(i)$ defines the rank of candidate i (lower is better). We also use $i \succ_{\sigma_k} j$ to denote $\sigma_k(i) < \sigma_k(j)$, i.e., i is ranked higher than j according to agent k. Furthermore, $|\sigma_k| \leq m$ is the number of candidates voted for by agent k. Note that $\sigma_k(j)$ is undefined for any candidate $j \in A \setminus A_k$. In such a case we say the vote is *incomplete*. By inserting remaining alternatives $A \setminus A_k$ in an incomplete ranking, we can construct a potential underlying complete ranking of all alternatives. This is called a *completion* (or extension [17]). Furthermore, we sometimes use $D = \{\sigma_1, \ldots, \sigma_n\}$ (for observed data) to denote all votes.

If we have access to the underlying true ranking, we can measure the quality of a voting rule on a given profile of votes by the distance of the aggregated ranking to the true ranking. The most common distance metric, and the one we use in this paper, is the Kendall-tau distance [16]. In detail, the Kendall-tau distance K counts the pairs of alternatives that are differently ordered by σ than by τ.

$$K(\sigma, \tau) = |\{\{i, j\} \subseteq A : i \succ_\sigma j \text{ and } i \prec_\tau j\}| \quad (1)$$

Such a differently ordered pair i and j is called an *inversion*. The Kendall-tau distance can be found in $\mathcal{O}(m \ln m)$ using a folk algorithm variant of merge sort called "count inversions".

The rule selecting an aggregate ranking τ^* which minimises the Kendall-tau distances to all votes is called the Kemeny optimal aggregation rule, and is given by:

$$\tau^* = \arg\min_\tau \sum_{k \in N} K(\sigma_k, \tau) \quad (2)$$

A more convenient way to write this is:

$$\tau^* = \arg\min_\tau \sum_{\{i,j\} \subseteq A : i \succ_\tau j} n_d(i, j|D), \quad (3)$$

[2]This differs from [31] who consider any partial order.

where $n_d(i, j|D) = |\{k \in N : i \prec_{\sigma_k} j\}|$ is the number of voters who disagree with the order $i \succ j$. Likewise, $n_a(i, j|D) = |\{k \in N : i \succ_{\sigma_k} j\}|$ denotes the number of agreements. Note that, in case of complete rankings, we have that $n_a(i, j|D) + n_d(i, j|D) = n$. However, this is not necessarily the case when rankings are incomplete.

We now describe the model for noisy and missing observations/votes. We assume noise according to the well-known Mallows' model for a probability $p > 0.5$. In this model, the likelihood of a ranking τ given observed votes D is:

$$\mathcal{L}(\tau|D) = \frac{1}{Z_1} \prod_{\{i,j\} \subseteq A : i \succ_\tau j} p^{n_a(i,j|D)} (1-p)^{n_d(i,j|D)}, \quad (4)$$

where Z_1 is a (normalisation) constant. It has been shown that Kemeny optimal chooses the ranking with the highest likelihood [32].

We extend this model to incomplete rankings by introducing the probability of a vote missing, where this probability is given by q. We assume that this probability is independent of the position in the true ranking. Incorporating this probability, we can compute the likelihood of a ranking τ given observations D as follows:

$$\mathcal{L}'(\tau|D) = \mathcal{L}(\tau|D) \frac{1}{Z_2} \prod_{k \in N} (1-q)^{|\sigma_k|} q^{m - |\sigma_k|}. \quad (5)$$

This assumption underlying this model has sometimes been called the "missing at random assumption" [14].

It is easy to see that Kemeny optimal still maximises the likelihood, irrespective of the value of q.

THEOREM 1. *Kemeny optimal is the maximum-likelihood estimator for Equation 5.*

PROOF. The ranking which maximises the likelihood is also maximising the log-likelihood. This allows us to drop all constants, including the normalisations Z_1 and Z_2 and even the incompleteness probabilities. Therefore:

$$\arg\max_\tau \mathcal{L}'(\tau, D) = \arg\max_\tau \ln(\mathcal{L}'(\tau, D)) =$$

$$\arg\max_\tau \sum_{\{i,j\} \subseteq A : i \succ_\tau j} (n_a(i, j|D) \ln(p) + n_d(i, j|D) \ln(1-p))$$

Since $p > 0.5$ we know that $\ln(p) > \ln(1-p)$, and because the total sum of ranked pairs is constant, we conclude that this log-likelihood is maximised if the number of agreements is (i.e., the number of disagreements is minimised). □

3. MINIMISING THE ERROR

So far we have discussed maximising the likelihood. Instead, the goal in this paper is to minimise the *rank aggregation error*, which we define as the *expected Kendall-tau distance*. Formally this is given by:

$$\text{KT-error}(\tau, D) = \sum_{\tau' \in T} K(\tau, \tau') \cdot \mathcal{L}'(\tau'|D), \quad (6)$$

where T is the set of all possible rankings.

Below we first discuss several examples to show that the two aims can result in different rankings, and then discuss the computational hardness of minimising the error and a local search approach for finding incremental improvements.

3.1 Likelihood vs. Error

Minimising the expected Kendall-tau distance and maximising the likelihood result in different rankings. We start by showing that this is true for instances with candidates that do not occur in any vote. We call such a candidate *free*.

DEFINITION 1. *A candidate $a \in A$ is* free *when none of the votes D contain a.*

EXAMPLE 1. *Let three candidates a, b, c be given, and one agent with vote $a \succ_{\sigma_1} b$. Kemeny's rule is indifferent between the three possible aggregate rankings (without noise (i.e., $p = 1$), each has a likelihood of $\frac{1}{3}$). The expected Kendall-tau distances for each of these, however, differ:*

τ_i	*ranking*	$\mathcal{L}(\tau_i\|D)$	$K(\tau_i,\tau_j)$			*KT-error*(τ_i, D)
τ_1	$a \succ b \succ c$	$\frac{1}{3}$	*0*	*1*	*2*	*1*
τ_2	$a \succ c \succ b$	$\frac{1}{3}$	*1*	*0*	*1*	$\frac{2}{3}$
τ_3	$c \succ a \succ b$	$\frac{1}{3}$	*2*	*1*	*0*	*1*

Because the distance of $a \succ c \succ b$ to each of the other rankings is only 1, it has a lower expected error. Note that the free candidate here is in the middle of the ranking.

This example shows that minimising the error produces a single natural ranking with the free candidate in the middle, where the likelihood is the same for multiple rankings. Generally:

PROPOSITION 1. *On an instance D with free candidates:*

1. *Kemeny's rule is indifferent between the position of free candidates (i.e., each position is equally likely);*
2. *The KT-error is minimised when free candidates are positioned in the median of the ranking.*

PROOF. Let a ranking π of m candidates be given. Let T be the set of all rankings of length m, $\mathcal{L}(\tau|D)$ denote the likelihood of a ranking $\tau \in T$, and let π_i (or τ_i) denote the ranking π (or τ, respectively) where the free candidate is placed at position $i \in \{0, \ldots, m\}$.

For the first statement, since the free candidate does not appear in D, each position is equally likely, and therefore, for any τ_j the likelihood $\mathcal{L}(\tau_j|D) = c \cdot \mathcal{L}(\tau|D)$, with $c = \frac{1}{m+1}$. Therefore, with Theorem 1, Kemeny's rule places free candidates at every position with equal probability.

For the second statement, we show that inserting a free candidate in the middle of the ranking minimises the expected error. By definition of the KT-error and $\mathcal{L}(\tau_j|D) = c \cdot \mathcal{L}(\tau|D)$ from above, we have that

$$\text{KT-error}(\pi_i, D) = \sum_j \sum_{\tau \in T} K(\pi_i, \tau_j) \cdot c \cdot \mathcal{L}(\tau|D).$$

The distance (error) of π_i to τ_j is equal to the distance of π to τ except for the difference in the position of the free candidate, i.e., $|i - j|$. Consequently,

$$\begin{aligned}&\text{KT-error}(\pi_i, D)\\ &\quad = c \cdot \textstyle\sum_j \sum_{\tau \in T} (K(\pi, \tau) + |i - j|) \cdot \mathcal{L}(\tau|D)\\ &\quad = \text{KT-error}(\pi, D) + c \cdot \textstyle\sum_j |i - j| \cdot \sum_{\tau \in T} \mathcal{L}(\tau|D)\end{aligned}$$

This is minimal if and only if $\sum_j |i - j| = \sum_{j=0}^{i-1}(i - j) + \sum_{j=i+1}^{m}(j - i)$. This is minimal for $i = \frac{m+1}{2}$. With induction this also holds for a set of free candidates. □

Next, we show that the difference between the two objectives goes even beyond free candidates, and can result in different rankings even when the votes are complete.

EXAMPLE 2. *Let $p = 0.7$ and five complete votes be given: twice $a \succ b \succ c$, twice $c \succ a \succ b$ and once $b \succ c \succ a$.*

τ_k	*ranking*	$\sum n_a$	$\sum n_d$	$\mathcal{L}(\tau_k\|D)$	*KT-error*
τ_0	$a \succ b \succ c$	*9*	*6*	*0.361*	*1.123*
τ_1	$a \succ c \succ b$	*6*	*9*	*0.028*	*1.877*
τ_2	$b \succ a \succ c$	*8*	*7*	*0.155*	*1.035*
τ_3	$b \succ c \succ a$	*7*	*8*	*0.066*	*1.965*
τ_4	$c \succ a \succ b$	*9*	*6*	*0.361*	*1.123*
τ_5	$c \succ b \succ a$	*6*	*9*	*0.028*	*1.877*

Then the Kemeny rule selects $a \succ b \succ c$ or $c \succ a \succ b$, but the ranking that minimises the KT-error is $b \succ a \succ c$.

Also here, we see that the MLE is not minimising the expected error.

3.2 Hardness

Finding the ranking with the largest likelihood (i.e., Kemeny optimal) is NP-complete [15]. Towards establishing the computational complexity of finding an aggregate ranking with the minimum error, we can show that the problem of computing the error of a (single) aggregate ranking is #P-hard (even) when there is no noise in the data D. The proof uses a reduction from computing the number of linear extensions of a partial order, which is #P-complete [2].

In the proofs below, let T_D denote the set of extensions of the partial order defined by the votes of D and $x_D = |T_D|$ the number of such extensions. For our proof we need the following lemmas.

LEMMA 1. *$\mathcal{L}(\tau|D)$ is the same for all consistent rankings $\tau \in T_D$ if D is generated from a model without noise, and this is equal to $\frac{1}{x_D}$.*

This follows because incompleteness is determined independently from the position in the ranking.

If D does not imply a complete linear order, there is a pair a, b of unordered alternatives. We express the number of consistent extensions of D in terms of the numbers of extensions for both possible orders for a and b as follows.

LEMMA 2. *Given a ranking π and data D without noise on a set of alternatives A. Let $a, b \in A$ be given. Let D_{ab} denote $D \cup \{(a, b)\}$. Then*

$$\begin{aligned}KT\text{-}error(\pi, D) \cdot x_D = \;& KT\text{-}error(\pi, D_{ab}) \cdot x_{D_{ab}}\\ +\; & KT\text{-}error(\pi, D_{ba}) \cdot (x_D - x_{D_{ab}}).\end{aligned}$$

PROOF. With Lemma 1, $\mathcal{L}(\tau|D) = \frac{1}{x_D}$. Applying the definition of KT-error, we write

$$\begin{aligned}&\text{KT-error}(\pi, D) \cdot x_D = \textstyle\sum_{\tau \in T_D} K(\pi, \tau)\\ &\quad = \textstyle\sum_{\tau \in T_{D_{ab}}} K(\pi, \tau) + \sum_{\tau \in T_{D_{ba}}} K(\pi, \tau).\end{aligned}$$

Using the definition of KT-error and of D_{ab} and D_{ba} the result then follows. □

The idea of the proof below is now to repeatedly use this fact that the number of consistent extensions of D is equal to the sum of the number of consistent extensions of $D \cup \{(a, b)\}$ and of $D \cup \{(b, a)\}$. By repeatedly adding (yet) unordered pairs of a and b to D, we collect a polynomial number (at most m^2) of linear constraints on the numbers of consistent extensions of increasingly larger sets of votes, ultimately leading to a single consistent extension.

THEOREM 2. *Given data D generated from a model without noise, determining the expected error of an aggregate ranking π is #P-hard.*

PROOF. We show this by a (polynomial) reduction from the problem of computing the number of linear extensions of a partial order $\succ$. Let such a partial order $\succ$ over a set of candidates A be given. First, for each pair of candidates (a, b) with $a \succ b$ in the partial order, insert an incomplete vote with a before b in D. Let a ranking $\pi \in T_D$ be given (e.g., by taking a topological order). Initially, D has x_0 consistent extensions. Then execute the following algorithm.

1. Initialise i to 0 and $e_0 = \text{KT-error}(\pi, D)$.
2. For every pair (a, b), if adding $a \succ b$ does not create a cycle in the majority graph of D then
   (a) Increment i, set $D_{ab} = D \cup \{(a, b)\}$ and $D_{ba} = D \cup \{(b, a)\}$.
   (b) Let $e_i = \text{KT-error}(\pi, D_{ab})$.
   (c) Let C_i denote the constraint $e_{i-1} \cdot x_{i-1} = e_i \cdot x_i + \text{KT-error}(\pi, D_{ba}) \cdot (x_{i-1} - x_i)$.
   (d) Let $D = D_{ab}$.

Let k denote i in the last iteration. All thus found constraints C_i are valid because of Lemma 2. When the last pair of candidates has been added to D, at index $k \leq m^2$, there is only one consistent extension, hence $x_k = 1$. If we know a value x_i, we can compute x_{i-1} using the equality constraint C_i with both x_i and x_{i-1}. So with induction we can compute x_0 in k such steps. This gives us the total number of consistent extensions of the partial order in polynomial time. Since with this polynomial number of calls to KT-error we solved a #P-hard problem, we conclude that computing KT-error is also #P-hard. □

3.3 Local Search

Although computing the error for candidate rankings to find the minimal one does not seem to be feasible, we can improve a ranking by making local adjustments. In particular, given a ranking τ, it is easy to determine if, by swapping two adjacent candidates, we can improve the KT-error.

THEOREM 3. *Let τ_{ab} and τ_{ba} be two equal rankings except that two adjacent candidates, a and b, are swapped. That is, $a \succ_{\tau_{ab}} b$ and $b \succ_{\tau_{ba}} a$. Then: KT-error$(\tau_{ab}, D) <$ KT-error(τ_{ba}, D) iff $n_a(a, b|D) > n_a(b, a|D)$.*

PROOF. Let $T_{ab} \subset T$ denote the set of all rankings where $a \succ b$ (not necessarily adjacent ones) and $T_{ba} = T \backslash T_{ab}$ the set of all rankings where $b \succ a$. Then $T_{ab} \cup T_{ba}$ is a partition of all possible rankings and $|T_{ab}| = |T_{ba}|$. We can thus write the KT-error(τ, D) (using Equation 6) as:

$$\sum_{\tau' \in T_{ab}} K(\tau, \tau') \cdot \mathcal{L}(\tau'|D) + \sum_{\tau' \in T_{ba}} K(\tau, \tau') \cdot \mathcal{L}(\tau'|D).$$

For $\tau \in T_{ab}$, from the definition of the Kendall-tau distance, we know that $K(\tau_{ba}, \tau) = K(\tau_{ab}, \tau) + 1$, and for $\tau \in T_{ba}$, $K(\tau_{ba}, \tau) = K(\tau_{ab}, \tau) - 1$. Therefore the KT-error$(\tau_{ba}, D)$ is

$$\begin{aligned}
&= \sum_{\tau \in T_{ab}} (K(\tau_{ab}, \tau) + 1) \cdot \mathcal{L}(\tau|D) \qquad (7)\\
&\quad + \sum_{\tau \in T_{ba}} (K(\tau_{ab}, \tau) - 1) \cdot \mathcal{L}(\tau|D)\\
&= \sum_{\tau \in T_{ab}} K(\tau_{ab}, \tau) \cdot \mathcal{L}(\tau|D) + \sum_{\tau \in T_{ab}} \mathcal{L}(\tau|D)\\
&\quad + \sum_{\tau \in T_{ba}} K(\tau_{ab}, \tau) \cdot \mathcal{L}(\tau|D) - \sum_{\tau \in T_{ba}} \mathcal{L}(\tau|D)\\
&= \text{KT-error}(\tau_{ab}, D) + \sum_{\tau \in T_{ab}} \mathcal{L}(\tau|D) - \sum_{\tau \in T_{ba}} \mathcal{L}(\tau|D).
\end{aligned}$$

Note that $\sum_{\tau \in T_{ab}} \mathcal{L}(\tau|D) + \sum_{\tau \in T_{ba}} \mathcal{L}(\tau|D) = 1$, or, more generally, a constant (normalisation is not relevant here).

We can express the likelihood that a comes before b given the data as:

$$\sum_{\tau \in T_{ab}} \mathcal{L}(\tau|D) = \mathcal{L}(a \succ b|D) = \frac{1}{Z} p^{n_a(a,b|D)} (1-p)^{n_a(b,a|D)}.$$

Using Equation 7 we thus can write the difference between errors, KT-error(τ_{ba}, D) − KT-error(τ_{ab}, D), as:

$$\frac{1}{Z} \left(p^{n_a(a,b|D)} (1-p)^{n_a(b,a|D)} - p^{n_a(b,a|D)} (1-p)^{n_a(a,b|D)} \right)$$

Since $p > 0.5$, this difference is strictly positive (negative) iff $p^{n_a(a,b|D)} > p^{n_a(b,a|D)}$ (or $p^{n_a(a,b|D)} < p^{n_a(b,a|D)}$). □

It turns out that repeatedly applying this rule until a local optimum is found has been called local Kemenisation [13]. It has been shown that any ranking thus produced satisfies the generalised Condorcet criterion (i.e., if there is a partition of the candidates (A_1, A_2) such that for every $a \in A_1$ and $b \in A_2$ the majority prefers a to b, then every $a \in A_1$ must be ranked above every $b \in A_2$ [28]). The above proof adds that this is locally minimising the KT-error as well. Note that although the proof assumes Mallows' model, it seems intuitive for any noise model to swap adjacent candidates if one is ranked more often above the other, and that this is independent of the (often unknown) value for p in the model.

4. EXPERIMENTS

We now empirically evaluate a range of voting rules, to determine their performance in settings with incomplete rankings. To this end, we first discuss the experimental setup and data, followed by the voting rules and modifications to deal with incomplete rankings. Then we discuss the results.

4.1 Setup and Data

We consider two types of experiments: those generated using synthetic data, and those based on real data from the PrefLib library [24], specifically the Mechanical Turk Dots and Puzzle experiments by [23]. We focus on these datasets since they provide noisy (albeit complete) rankings and they also include an objective ground truth. In more detail, for the synthetic data, we use the repeated insertion method discussed in [10, 20] to generate noisy rankings according to the Mallows' model. Furthermore, to generate incomplete rankings, we independently remove each candidate from each agent with a probability q.

The real data consists of several datasets, each containing up to 800 agents ranking 4 candidates. In the Dots experiment, each voter was asked to rank 4 images according to the number of dots they contained, whereas in the Puzzle game the voters were asked to rank sliding puzzles according to how close they were to the solution (see [23] for details). The votes contain natural noise and are complete. To make the votes incomplete, we remove each candidate from each agent with probability q as before. In addition, we randomly select n agents without replacement (where we vary n).

We repeat each experiment 1000 times with resampled random values for candidates to determine the order in case of ties in the scoring rules, and we measure the average Kendall-tau distance (i.e., the number of inversions) of the aggregated ranking to the true ranking. Note that this is consistent with our objective of minimising the KT-error.

4.2 Voting Rules for Incomplete Votes

We consider the following common voting rules in the literature on rank aggregation. For each of the rules below, we also add a variant with local search, where we improve the rank produced by the corresponding rule by applying the algorithm described in Section 3.3 until it converges.

Borda.

According to the Borda rule every agent assigns $n-j$ points to the candidate ranked in position j, which is equal to the number of candidates it defeats. Candidates are then ranked according to the sum of points for each candidate, which is also called the *Borda count*. However, with incomplete votes, it is not clear how many points should be awarded to the candidates ranked by an agent and the ones missing. There are many variants (see, e.g., [1, 26, 13]) and we consider the following three: Pessimistic, where ranked candidates contribute a score of $(m-\sigma_k(i))$ and unranked ones contribute zero points; Optimistic, which is the same except that unranked candidates contribute $(m-|\sigma_k|)$; Scaled, where the score is proportional to the position within the ranked candidates, $m(|A_k|-\sigma_k(i))/|A_k|$, and missing candidates contribute zero. We choose these three since they vary widely in their performance, whereas other variants we tried performed similarly to one of these three.

Spearman's Footrule.

Spearman's footrule is another commonly-analysed voting rule, especially for the Mallows' noise model, since it is a 2-approximation of Kemeny optimal but computable in polynomial time. This rule minimises the sum of Spearman's distances of the complete ranking to the votes, where the distance between two rankings σ and τ is given by $S(\sigma,\tau)=\sum_{i\in A}|\sigma(i)-\tau(i)|$. For complete rankings, this is done by finding the minimal weighted matching of alternatives to their ranks in the aggregate ranking, where the weight w_{ij} of a candidate i in position j is given by $w_{ij}=\sum_k|\sigma_k(i)-j|$ (computable in $\mathcal{O}\left(m^3\right)$ using the Hungarian algorithm). Now, as with Borda, there are several ways to extend this rule to deal with incomplete votes. We choose *Scaled Footrule Optimal* (SFO) from [13] since it is simple and computationally tractable (unlike, e.g., using the induced distance [13]). In detail, to compute the distance for candidate i at position j, instead of using $\sigma_k(i)$ and j, both of these are scaled according to the total number of candidates. Specifically, the weight is replaced by $w_{ij}=\sum_k|\sigma_k(i)/|A_k|-j/m|$. This formulation represents the idea that the missing alternatives are equally spread in between the ranked alternatives.

Copeland.

The Copeland voting rule ranks individual candidates according to the number of wins in pair-wise contests minus the number of losses. This rule can be readily applied to incomplete settings by only counting pairs when both alternatives appear in an agent's ranking. Formally, let $P(i,j)=|\{k\in A: i,j\in A_k\wedge i\succ_{\sigma_k} j\}|$ denote the number of agents who prefer i to j. Then, candidates i's score is computed by:

$$|\{j\neq i: P(i,j)>P(j,i)\}|-|\{j\neq i: P(i,j)<P(j,i)\}|$$

Candidates then are ranked according to their score in descending order.

Kemeny optimal.

We implement the Kemeny optimal rule by a mixed integer optimisation problem on the weighted majority graph [7] for which we (uniform) randomly select one optimal solution.

Optimal.

Computing the KT-error exactly is hard (Theorem 2), but in practice we can still compute the Optimal for up to 6 candidates using a brute force approach. Specifically, the KT-error is computed for all possible rankings and then the one with the minimal KT-error is chosen. Note, however, that the definition of KT-error for Mallows' model depends on the noise probability p. For the synthetic experiments we simply use the p value that was used for generating the instances. For the experiments with the real data we compute the KT-error for a range of p values to establish the best one experimentally. As before, if there are multiple optimal solutions, we select one randomly.

4.3 Results

Figure 1 shows results using synthetic data with 6 candidates, a Mallows noise probability $p=\frac{2}{3}$ and a probability of candidates missing of $q=0.7$ for different values of the number of agents. The right figure shows the results after applying local Kemenisation to each of the voting rules. As expected, having more agents decreases the average distance to the true ranking for all rules. We also can observe that Kemeny is indeed not optimal, with on average around 0.5 inversions more than Optimal. Interestingly, Copeland performs significantly better than Kemeny, and at times on par with Optimal. Even more striking is the significant improvement of most rules by local Kemenisation, which can be observed by comparing the left to the right figure. We have similar results for other values for q, but show only $q=0.7$, because here the differences are most pronounced.

This can be seen in Figure 2, where we vary the probability q of missing candidates and show the average distance for all rules for a scenario with $p=\frac{2}{3}$ and 25 agents. Similar to the previous results, we see that Copeland consistently outperforms Kemeny, and that Kemeny is relatively far from optimal. Compared to the previous figure, we see here that differences between the rules are less pronounced for lower values of q. In particular, after applying local Kemenisation, none of the rules are statistically different up to $q=0.6$.

The results so far have considered data where the synthetic noise model is consistent with our objective. We now consider the real data, which uses natural noise generated through experiments rather than a particular model. To this end, Figures 3 and 4 show the results from the Dots dataset (number 1) and Puzzle (number 2) respectively, where we vary the number of agents and set the probability of missing a candidate to $q=0.7$. Surprisingly, trends for both datasets are very similar to the synthetic data: despite the fact that Optimal is not necessarily optimal with real data (since it assumes the Mallows' model), it significantly outperforms all other voting rules. Furthermore, Copeland outperforms Kemeny in most instances. Finally, again despite the fact that it assumes Mallows' model, local Kemenisation significantly improves most voting rules, except of course Kemeny and Optimal, which are already locally optimal, and Copeland for some instances (within the standard error). We can see the same trend for the other Dots and Puzzle instances (results not shown).

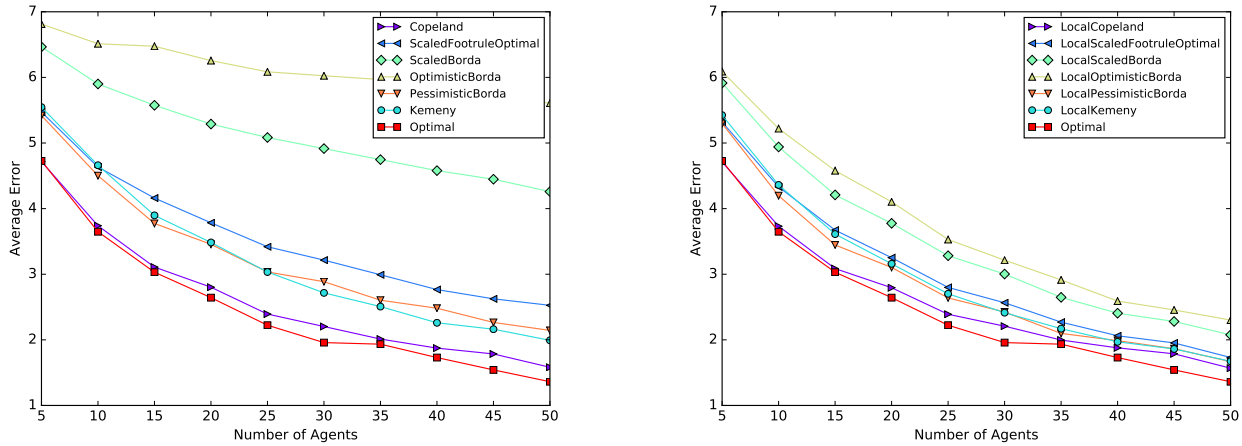

Figure 1: More agents decrease the average distance for all rules (6 candidates, $p = \frac{2}{3}$, and $q = 0.7$). Copeland performs better than Kemeny, and local Kemenisation (right) significantly improves most other rules.

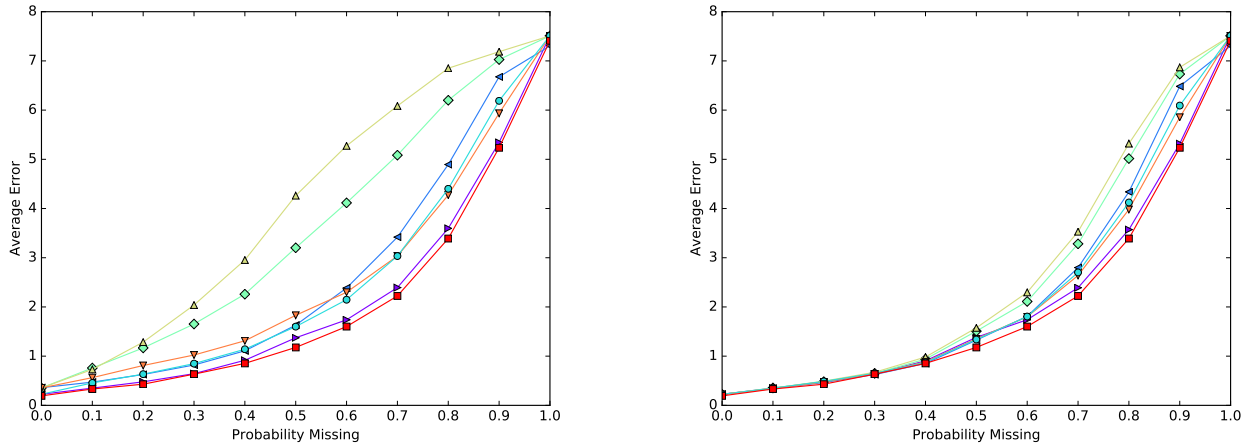

Figure 2: Missing candidates increase the average distance for all rules (6 candidates, $p = \frac{2}{3}$, and 25 agents).

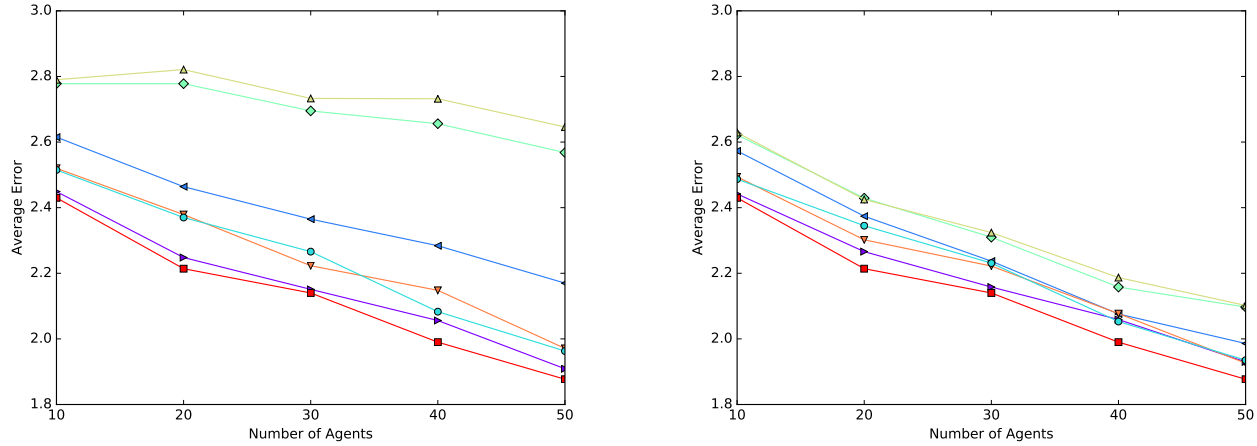

Figure 3: The relative performance of the rules on the Dots data set 1 with a probability of removing a candidate of 0.7 is similar to the synthetic data.

Table 1: The average distances for Optimal for different values of p are given on the Dots 1 problem instance for 10 agents (top) and 50 agents (bottom) and for probabilities 0.0–1.0 of missing candidates. Except for $p = 0.5$ these differences are statistically insignificant (standard errors of above 0.02).

p	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.5000	3.004	3.03	2.964	3.063	2.998	3.002	2.974	2.94	2.984	2.946	3.055
0.6225	1.615	1.679	1.815	1.965	2.054	2.192	2.315	2.446	2.691	2.815	3.055
0.7311	1.606	1.683	1.822	1.991	2.059	2.203	2.314	2.435	2.681	2.844	3.055
0.8176	1.605	1.685	1.82	1.989	2.057	2.207	2.335	2.439	2.705	2.81	3.055
0.8808	1.607	1.684	1.819	1.992	2.059	2.212	2.326	2.43	2.694	2.815	3.055
0.9241	1.611	1.69	1.824	1.997	2.061	2.208	2.346	2.447	2.68	2.813	3.055
0.9526	1.611	1.676	1.824	1.989	2.055	2.216	2.323	2.451	2.703	2.816	3.055
0.5000	2.946	2.936	3.006	3.083	3.023	2.978	3.016	2.986	3.069	3.125	2.964
0.6225	0.733	0.834	0.89	1.006	1.274	1.405	1.597	1.887	2.2	2.684	2.964
0.7311	0.731	0.833	0.898	1.02	1.285	1.429	1.627	1.877	2.185	2.67	2.964
0.8176	0.733	0.834	0.9	1.023	1.285	1.424	1.629	1.884	2.207	2.66	2.964
0.8808	0.734	0.834	0.9	1.023	1.284	1.427	1.628	1.89	2.189	2.7	2.964
0.9241	0.732	0.832	0.899	1.025	1.284	1.424	1.626	1.88	2.186	2.71	2.964
0.9526	0.731	0.834	0.901	1.024	1.284	1.429	1.627	1.885	2.197	2.653	2.964

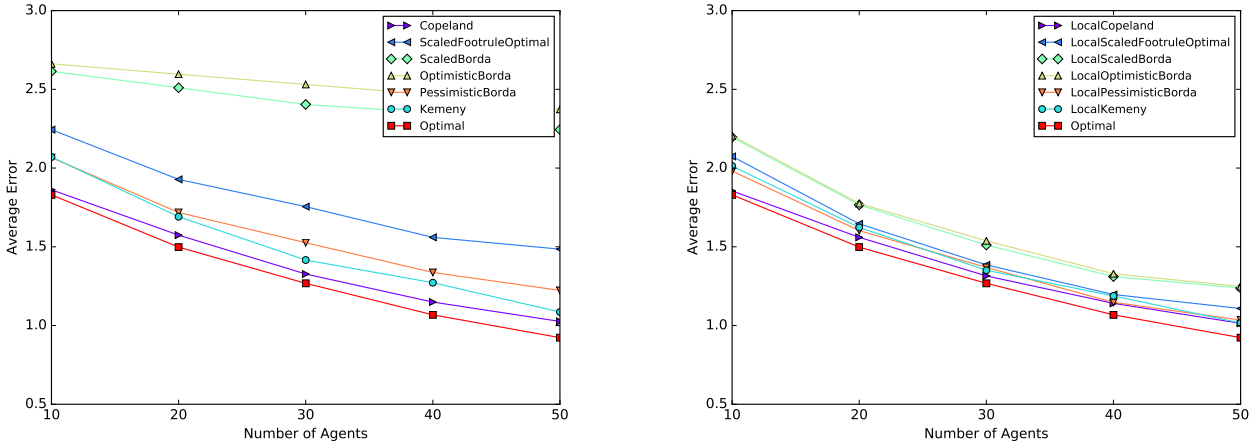

Figure 4: The performance of the rules on this Puzzle data set 2 with $q = 0.7$ shows the same trends as for Dots.

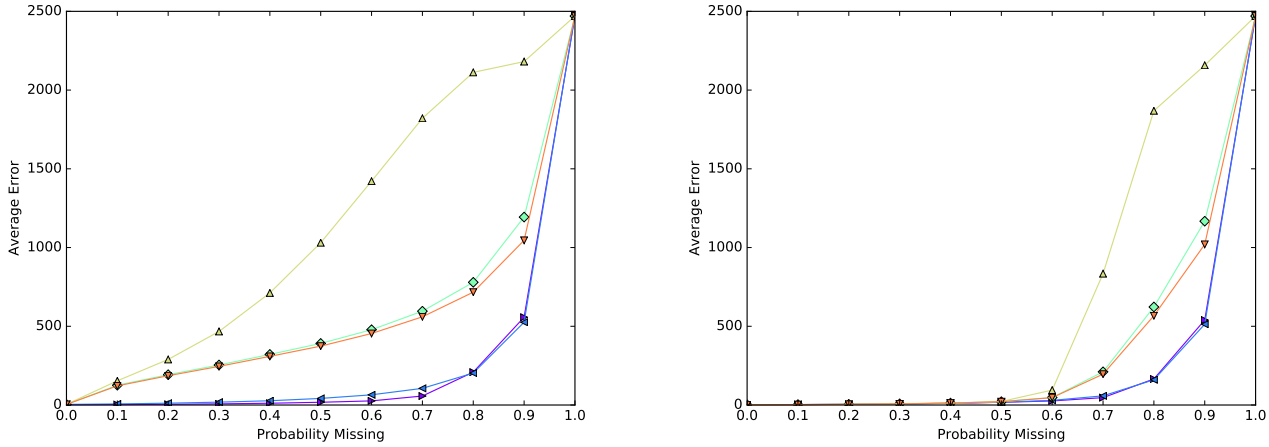

Figure 5: Local Kemenisation reduces the error drastically with 100 candidates, 40 agents and $p = \frac{2}{3}$.

We furthermore considered the effect of the unknown noise probability p on the performance of Optimal with real data. The results are shown in Table 1 for a range of q and p values, where the p values are computed using a generalised noise model [12] $p = \frac{e^\alpha}{1+e^\alpha}$ and $\alpha \in \{0, 0.5, \ldots, 3\}$. Interestingly, the results are not statistically significant for different values of p (except 0.5), suggesting that Optimal is robust with respect to the choice of p.

We also produced synthetic experiments for larger numbers of candidates (100) and more agents (up to 200). Although we were unable to compute Kemeny and Optimal rankings for these settings, the trends for the remaining voting rules were largely the same. A representative example in shown in Figure 5. As before, local Kemenisation significantly reduces the error.

5. RELATED WORK

The average Kendall-tau distance (among other measures) is often used in experiments with noisy and incomplete data to compare the effectiveness of a set of rules over a large set of problem instances, e.g., [11, 13]. However, this is typically done implicitly and without noticing that minimising this error is a significantly different objective from maximising the likelihood. The objective of minimising the error is first explicitly mentioned in a technical report [29, Section 5]. They say it is (more) difficult to optimise for this than maximising the likelihood (without proof) as "no closed form solution exists", which is in line with our hardness result. Later this statistical decision-theoretic viewpoint on social choice and the hardness proof have been formalised [27], but only considers complete votes.

A related argument against using maximum likelihood as the objective is that optimising for a single noise model may not be optimal in realistic settings, because the noise could take unpredictable forms [4, 23]. This has led to the design of a "modal" ranking rule that is robust against any "reasonable" noise model [4]. Following a similar argument, it is relevant to learn a mixture of (Mallows) noise models, e.g., through a Monte Carlo based approximation [21]. Such approaches can be seen as complementary to the direction we take in this paper. In fact, we argue that the ultimate objective should be to minimise the error based on a learned mixture of general noise models.

Specifically regarding incomplete votes, some work has considered machine learning techniques that view the missing ranking information as hidden variables, which are then inferred from other votes. For example, Cheng et al. [6] use the Expectation–Maximisation algorithm. Other experiments with real data show the performance of a number of "existing, standard, algorithms from machine learning" to infer the missing information [11]. For an overview of the workflow of designing social choice mechanisms using machine learning see the paper by Xia [30].

Much of the work in this area uses information retrieval as its main application domain (e.g., [13, 19]) and machine learning is used as an adaptive voting rule which learns how to rank the documents. An important challenge in this domain is scalability, especially for search engines, where the number of candidates (documents) can be as large as several billions. So far, this problem has mostly been approached as a machine learning classification task. However, other voting rules such as SFO, Borda, and a range of methods using a Markov decision model of the votes, have been evaluated on a web page data set [13]. Consistent with our findings, their results show that the rule said to be similar to Copeland (called MC_4) performs the best on this set. Importantly, however, the "ground truth" in such applications is not a full rank, but rather whether a document is relevant or not. As a result, the objective in those approaches is different (they are more concerned with measures such as recall and precision and other measures specifically relevant to information retrieval). Nevertheless, our results support the main conclusions from these papers in that the Copeland rule seems an appropriate choice for most levels of noise and missing candidates.

6. CONCLUSIONS

We have shown that voting rules which maximise the likelihood of a ranking do not necessarily minimise the rank aggregation error, i.e., the expected distance to the true ranking. Specifically, for rank aggregation with significant noise and missing votes, maximising the likelihood (i.e., using Kemeny's rule assuming Mallows' model) can result in a significantly higher error than computationally simpler methods such as Copeland. While the results are particularly pronounced with missing votes, we have shown that this discrepancy can occur even when votes are complete. Furthermore, we have shown that Optimal performs best in both synthetic and real data settings, even when we do not know the noise parameter exactly. In terms of theoretical results, for Mallows' model we have shown that computing this error is hard. Furthermore, we proved that an efficient procedure called local Kemenisation, which is known to improve the likelihood, also reduces the error, and that in fact this leads to a significant performance improvement for varying incompleteness and noise levels.

The next logical step is to design new voting rules with the objective of minimising the error in settings with incomplete and noisy observations. This would be particularly interesting for more general (mixtures of) noise models. These extensions also give rise to a number of questions regarding the complexity class of the problems of minimising the rank aggregation error. In particular, although we showed that computing the error is #P-hard, determining whether the complexity of finding the ranking with minimal error for the Mallows' model is also #P-hard is still an open problem. Other extensions include considering different incompleteness models (e.g., where the probability of missing depends on the position in the true ranking) and different distance measures (e.g., winner determination, top-k, or more general weighted measures). Additionally, it would be interesting to compare existing voting rules to approaches that apply machine learning methods, both through learning missing data [11], but also by directly applying classifiers as is common in the "learning to rank" information retrieval field [19].

Acknowledgments

This work has been partly supported by COST Action IC1205 on Computational Social Choice.

REFERENCES

[1] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 577–584, 2012.

[2] G. Brightwell and P. Winkler. Counting linear extensions is #P-complete. In *23rd ACM Symposium on Theory of Computation*, pages 175–181, 1991.

[3] I. Caragiannis, G. A. Krimpas, and A. A. Voudouris. Aggregating partial rankings with applications to peer grading in massive online open courses. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 675–683, 2015.

[4] I. Caragiannis, A. D. Procaccia, and N. Shah. When do noisy votes reveal the truth? In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 143–160. ACM, 2013.

[5] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202. ACM, 2013.

[6] W. Cheng, J. Hühn, and E. Hüllermeier. Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 161–168. ACM, 2009.

[7] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *AAAI*, volume 6, pages 620–626, 2006.

[8] V. Conitzer, M. Rognlie, and L. Xia. Preference Functions that Score Rankings and Maximum Likelihood Estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 9, pages 109–115, 2009.

[9] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2005.

[10] J.-P. . P. Doignon, A. Pekeč, and M. Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69(1):33–54, 2004.

[11] J. A. Doucette, K. Larson, and R. Cohen. Conventional Machine Learning for Social Choice. In *Proceedings of the 29th AAAI Conference on Artificial Intellligence*, 2015.

[12] M. Drissi-Bakhkhat and M. Truchon. Maximum likelihood approach to vote aggregation with variable probabilities. *Social Choice and Welfare*, 23(2):161–185, 2004.

[13] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *10th Int. Conf. on World Wide Web*, pages 613–622. ACM, 2001.

[14] Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical report, Massachusetts Institute of Technology, 1995.

[15] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[16] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[17] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.

[18] A. Kumar and M. Lease. Learning to rank from a noisy crowd. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1221–1222. ACM, 2011.

[19] T.-Y. . Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[20] T. Lu and C. Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 145–152, 2011.

[21] T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *The Journal of Machine Learning Research*, 15(1):3783–3829, 2014.

[22] C. L. Mallows. Non-null ranking models. I. *Biometrika*, pages 114–130, 1957.

[23] A. Mao, A. Procaccia, and Y. Chen. Better Human Computation Through Principled Voting. In *AAAI*, pages 1142–1148, 2013.

[24] N. Mattei and T. Walsh. PrefLib: A library of Preference Data. In *Algorithmic Decision Theory*, pages 259–270. Springer, 2013.

[25] M. Nicolas de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'imprimerie royale, 1785.

[26] D. G. Saari. Mathematical structure of voting paradoxes: I. Pairwise votes. *Economic Theory*, 15(1):1–53, 2000.

[27] H. A. Soufiani, D. C. Parkes, and L. Xia. A statistical decision-theoretic framework for social choice. In *Advances in Neural Information Processing Systems*, pages 3185–3193, 2014.

[28] M. Truchon. An extension of the Condorcet criterion and Kemeny orders. Technical report, 1998.

[29] K. Tsukida and M. R. Gupta. How to analyze paired comparison data. Technical report, DTIC Document, 2011.

[30] L. Xia. Designing social choice mechanisms using machine learning. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pages 471–474, 2013.

[31] L. Xia and V. Conitzer. A maximum likelihood approach towards aggregating partial orders. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pages 446–451, 2011.

[32] P. Young. Optimal voting rules. *The Journal of Economic Perspectives*, 9(1):51–64, 1995.