Algorithmica (2010) 57: 284–300
DOI 10.1007/s00453-008-9211-1

Aggregation of Partial Rankings, p-Ratings and Top-m Lists

Nir Ailon

Received: 19 October 2007 / Accepted: 11 June 2008 / Published online: 16 July 2008

Abstract We study the problem of aggregating partial rankings. This problem is motivated by applications such as meta-searching and information retrieval, search engine spam fighting, e-commerce, learning from experts, analysis of population preference sampling, committee decision making and more. We improve recent constant factor approximation algorithms for aggregation of full rankings and generalize them to partial rankings. Our algorithms improve constant factor approximation with respect to a family of metrics recently proposed in the context of comparing partial rankings. We pay special attention to two important types of partial rankings: the well-known top-m lists and the more general p-ratings which we define. We provide first evidence for hardness of aggregating them for constant m, p.

Keywords Rank aggregation · Ranking with ties · Approximation algorithms

1 Introduction

Rank aggregation (see [3, 7, 10, 12–17] and references therein) is the problem of finding a ranking (permutation) π of a ground set V of n elements combining information from a list $\pi_1, \ldots, \pi_k$ of input rankings (*votes*). This problem is motivated by many applications such as meta-searching and information retrieval, search engine spam fighting, e-commerce, learning from experts, analysis of population preference

Most of this work done while author was a student in the Department of Computer Science at Princeton University, and part while a member of the Institute for Advanced Study, supported by the National Science Foundation under agreement No. DMS-0111298. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. A preliminary version appeared in [1].

N. Ailon (✉)
Google Research, 76 9th Avenue, New York, NY 10011, USA
e-mail: nailon@google.com

sampling, committee decision making and more. In addition to the practical motivation, there is a long history of theoretical interest in the mathematics arising from the problem (some classic milestones are [6, 8, 9, 21, 22]). For a nice survey, refer to [20].

Partial Rankings

In this work we consider *rankings with ties*, or, following more common terminology, *partial rankings*.[1] A partial ranking can be defined as a mapping π from V to any totally ordered universe U, which we call the *rank universe*. If $\pi(u) = \pi(v)$ then we say that u, v are in a *tie*. The objective functions we define below will be independent of the actual choice of the rank universe. Our approach is *comparison* based (in contrast with *score* based which we do not consider here). Hence, a partial ranking is characterized by the matrix of comparisons between $\pi(u)$ and $\pi(v)$ for $u, v \in V$. Using pairwise information has many advantages in problems related to both ranking and clustering [2, 18, 19, 25, 27].

Partial rankings arise in many natural situations, and are a natural enrichment of any comparison based system. As an example, Condorcet [9] considered election systems allowing voters to express full rankings of a set of candidates. By allowing ties, neutrality can be expressed with respect to pairs of candidates. Partial rankings are hence a natural extension of this classic theory.

In this work we extend work on rank aggregation to partial rankings. We motivate and pay particular attention to two restricted classes of partial rankings: *top-m lists* and *p-ratings*. The two classes arise ubiquitously in information retrieval systems and on the internet.

Top-m Rankings

One of the main drawbacks of considering *full rankings* is that expecting full ranking information of V from all voters can be too much to ask for [12–16]. In the search engine example, it is unlikely that a search engine would provide a ranking of the entire set V of all web pages matching a given query. Instead, only the first $m \ll n$ top-ranked pages are returned. If we denote the full ranking (implicitly) computed by the search engine by $\pi : V \to \{1..n\}$ ($\pi(v)$ is the *rank* of v, smaller numbers meaning further "ahead" in the list), then the search engine returns only $\pi^{-1}(1), \ldots, \pi^{-1}(m)$ to the client. Lacking rank information among the elements $\pi^{-1}(m+1), \ldots, \pi^{-1}(n)$, the next best option is to assume they belong to one big *tie*, which is itself ranked in a position $m + 1$.[2]

p-Ratings

Another important case of partial rankings are *ratings*. We define a p-rating to be a mapping from the ground set V to a rank universe U of size p for some fixed p (we

[1] The term 'partial ranking' used here should not be confused with two other standard objects: (1) *Partial order*, namely, a reflexive, transitive anti-symmetric binary relation; and (2) A ranking of a *subset*.

[2] The choice of $m + 1$ is a convenience. Any number greater than m will do. In a comparison based system the magnitude of the ranks is immaterial.

may assume that $U = \{1, \ldots, p\}$). Some good examples are: (i) Each hotel in a set V of hotels is rated as $\star$, $\star\star$, $\star\star\star$, $\star\star\star\star$, or $\star\star\star\star\star$ by hotel critics ($p = 5$). (ii) Financial experts advise to either *sell, hold* or *buy* each stock in some set V ($p = 3$). (iii) A company identifies its strengths and weaknesses in customer service by distributing questionnaires to its clients. The questionnaire is a table with rows (V) corresponding to different customer service aspects, and columns (U) correspond to an ordered range of p satisfaction levels. Each participating client marks an '$\times$' in a single box in each row, corresponding to their opinion.[3] Two elements in V are tied by a client if they marked an '$\times$' in the same column for the two elements. Thus, each client gives rise to a partial ranking. The company wants to sort the customer service aspects from strong to weak by aggregating the responses.

Note that top-m lists are a special case of $(m + 1)$-ratings, in which the preimage of all but the least preferred rank universe element are singletons.

The Problem

Partial rank aggregation is defined as the problem of aggregating a finite set of partial rankings over the same set V in a meaningful way. In this work the aggregated output σ is a full ranking of V. This choice (as opposed to outputting a partial ranking) allows expressing more information. Indeed, suppose $V = \{A, B, C\}$, and we wish to aggregate an input consisting of a list of ratings, where the rank universe is $U = \{good, bad\}$ (with $good < bad$). There are two voters, where the first rates A, B as *good* and C as *bad*, and the second rates A as *good* and B, C as *bad*. We write $\pi_1 = [AB, C]$ and $\pi_2 = [A, BC]$ as shorthand for this voting outcome. Although individual votes allow ties, the overall information in the votes breaks all ties. Outputting $[A, B, C]$ (i.e. a full ranking) captures all pairwise comparison information for this input. Degenerate cases in which A and B are tied in all input partial rankings will require special attention.

The objective cost function is based on a measure of distance d between partial rankings. Here we will mostly use the distance between full rankings (the output) and partial rankings (the input). Our distance measure generalizes the Kendall-τ measure [23] (originally defined as a *metric* on full rankings). The distance $d(\sigma, \pi)$ between partial rankings σ, π is the number of distinct $u, v \in V$ such that u is ranked strictly ahead of v in π and v is ranked strictly ahead of u in σ. Our goal is to minimize the sum of distances between the output and the individual partial rankings. This is the *Kemeny* approach to the aggregation problem and is considered to have many advantages [12, 13].

1.1 Previous Work

Fagin, Kumar, Mahdian, Sivakumar and Vee [14–16] provide a comprehensive picture on how to compare partial rankings. There, different natural measures of distance between partial rankings are suggested. These measures extend the well known

[3] We assume no column is labeled 'not applicable'.

Kendall-τ and Spearman's footrule [11] metrics on full rankings. Their main contribution is in showing that all the extensions they study belong to a class $\mathcal{D}$ of metrics that are equivalent up to global constants, and hence by optimizing or approximating with respect to one we approximate with respect to all. The measure of distance d suggested in this work also appears in their study. However, it does not belong to class $\mathcal{D}$. In fact, it is not a proper distance function, because $d(\sigma, \pi)$ may be zero for two distinct σ, π. In spite of the apparent difficulty in using an improper distance function, our results imply approximation algorithms with respect to all metrics in class $\mathcal{D}$ as well (shown in Sect. 6).[4]

This work combines algorithmic techniques from Ailon, Charikar and Newman's recent work on full rank aggregation (aggregation of full rankings) [4] with Fagin et al.'s aforementioned work. Our main results are two approximation algorithms for aggregating partial rankings. The first (Sect. 3) is new a 2-approximation, generalizing a well-known [3] 2-approximation for full rank aggregation. We then present a new 3/2-approximation algorithm in Sect. 4, generalizing a recent algorithm [3] for full rank aggregation to the problem of partial rank aggregation. In addition to showing the applicability of the previous algorithm to the domain of partial rankings, we improve it using a new technique of perturbing the variables of an optimal solution to an LP relaxation before rounding it. In Sect. 6 we show that our algorithms also imply constant factor approximation with respect to the class $\mathcal{D}$ of metrics discussed in [14, 15]. There a 3-approximation algorithm is suggested, but with respect to an objective function we do not use here.

Dwork et al. successfully experiment with several heuristics in [12, 13], some based on Markov chains. They suggest a general scheme for aggregating partial rankings where a greedy post-processing step (called local Kemenization) is applied to the output of any algorithm. Local Kemenization is shown to have many nice properties, and it could easily be applied as a final step for our algorithms.

Finally (Sect. 5) we show that aggregation of partial rankings is in P when the rank universe U consists of 2 elements, but becomes NP-hard already when it consists of 3 elements (even for the special case of aggregation of top-2 lists). It was previously known that aggregation of partial rankings is NP-hard because rank aggregation is a special case (shown to be NP-Hard by Dwork et al. [13]). Our result shows that the seemingly easier interesting cases of p-rating and top-m aggregation are already NP-Hard for extremely small rank universe (rank aggregation is *not* a special case of these problems). As far as we know, we are the first to define the problem of aggregating p-ratings (as a natural generalization of top-m lists) in the context of aggregating partial rankings and to provide evidence to both problems' hardness for constant m, p.

2 Definitions

We use $[n]$ to denote the integer set $\{1, \ldots, n\}$ equipped with the induced integer total order. We assume V is some ground set of n elements. Let E denote the set $\binom{V}{2}$ of all unordered pairs $\{u, v\} \subseteq V$ and T the set $\binom{V}{3}$ of all unordered triplets $\{u, v, y\} \subseteq V$.

[4] Under our restriction of outputting full rankings only.

Definition 1 A *partial-ranking* of V is a mapping $\pi : V \to U$ for some *rank universe* U equipped with a total order relation. For a partial ranking π, let $\pi(V)$ denote its image. Two partial-rankings $\pi : V \to U$, $\pi' : V \to U'$ of V are rank-equivalent if $\pi' = f \circ \pi$ for some strictly monotone $f : \pi(V) \to \pi'(V)$ (hence $\pi = f^{-1} \circ \pi'$). A *full-ranking* of V is a partial-ranking that is an injection. A *strict* partial-ranking is a partial-ranking that is not a full-ranking. A partial ranking $\pi : V \to U$ of V is a *p-rating* if $|U| = p$. A partial ranking π of V is a *top-m-ranking* (for $m \leq n$) if it is an $(m+1)$-rating, and the preimages of all but the maximal element of U are singletons. The trivial partial-ranking $\mathbf{1}_V$ is the constant mapping.

(The definition of a p-rating was inspired by an example suggested by Moses Charikar.) For the sake of computation, we can always assume $U = [n]$. Given a partial-ranking $\pi : V \to U$, we write $u <_\pi v$ for $u, v \in V$ if $\pi(u) < \pi(v)$. We write $u =_\pi v$ if $\pi(u) = \pi(v)$. Similarly, we define $>_\pi$, $\leq_\pi$ and $\geq_\pi$.

We now define a measure of distance between a full-ranking and a partial-ranking.

Definition 2 Given two partial-rankings σ and π of V, the generalized Kendall-τ distance $d(\sigma, \pi)$ between the two is defined as the number of ordered pairs $u, v \in V$ such that $u <_\sigma v$ and $v <_\pi u$.

Note that we will only need the distance between a full-ranking σ (the output) and a partial-ranking π (the votes). The distance between any partial-ranking and $\mathbf{1}_V$ is 0. Also note that d is not even a pseudometric, because $d([A, B], [AB]) = 0$, $d([B, A], [AB]) = 0$ and $d([A, B], [B, A]) = 1$, violating the triangle inequality. However, d restricted to full-rankings is a metric. In fact, an alternative way to define $d(\sigma, \pi)$ for a full ranking σ and partial ranking π is $d(\sigma, \pi) = \min_{\tau \in \Pi(\pi)} d(\sigma, \tau)$, where $\Pi(\pi)$ is the set of all full rankings consistent with π (we will not use this fact here).

We are now ready to define the optimization problems considered in this work.

Definition 3 PARTRANKAGG is the problem of, given a list $\pi_1, \ldots, \pi_k$ of partial-rankings of V (votes), outputting a full-ranking π minimizing $\text{cost}(\pi) = \frac{1}{k} \sum_{i=1}^{k} d(\pi, \pi_i)$. RANKAGG is PARTRANKAGG with the restriction that the votes are full-rankings. pRATINGAGG is PARTRANKAGG with the restriction that the votes are p-ratings. TOPmAGG is PARTRANKAGG with the restriction that each vote π_i is a top-m_i-ranking for some $m_i \leq m$.

Given an input $\pi_1, \ldots, \pi_k$ to PARTRANKAGG and distinct $u, v \in V$, we say that $u \equiv v$ if for all $i = 1, \ldots, k$, $u =_{\pi_i} v$. We define the *pairwise weight* $w_{uv} = \frac{1}{k} |\{i : u <_{\pi_i} v\}|$. Clearly (i) $w_{uv} + w_{vu} \leq 1$ for all u, v; (ii) $u \equiv v \Leftrightarrow w_{uv} = w_{vu} = 0$; (iii) For any full-ranking σ of V, $\text{cost}(\sigma) = \sum_{u <_\sigma v} w_{vu}$.

3 A 2-Approximation Algorithm for PARTRANKAGG

It is well known that RANKAGG admits a very simple randomized 2-approximation algorithm (called pick-a-perm in [3]): simply output a choice of $\pi_1, \ldots, \pi_k$ uniformly

```
REPEATCHOICE (V, π_1, ..., π_k)

set π ← 1_V
set σ ← arbitrary ranking of V
set S ← ∅
while ∃u, v s.t. u =_π v and u ≢ v
    choose i ∈ [k] \ S uniformly at random
    set π ← π_i * π
    set S ← S ∪ {i}
return σ * π
```

Fig. 1 Pseudocode for REPEATCHOICE

at random. On expectation, such a choice has cost at most twice the optimal solution. One way of proving this is by arguing that d is a metric when restricted to the space of full-rankings. The 2-approximation argument easily follows from this fact. In our case, d is not a metric, and moreover, it is not clear how to turn some vote π_i (which could be a strict partial-ranking) into a full-ranking.

To remedy these problems, we define a refinement operator between partial rankings (previously used in [14]).[5]

Definition 4 A refinement $\pi' * \pi$ of π by π' is the unique (up to rank-equivalence) partial-ranking σ satisfying

$$u <_\sigma v \quad \Longleftrightarrow \quad u <_\pi v \quad \text{or} \quad (u =_\pi v \text{ and } u <_{\pi'} v).$$

The following facts are immediate to verify:

- The element $\mathbf{1}_V$ is the identity with respect to $*$,
- The refinement operator $*$ is associative, and,
- The refinement $\pi' * \pi$ can be computed in polynomial time.

Algorithm REPEATCHOICE (Fig. 1) repeatedly chooses a random vote π_i (without repetitions) and refines the current partial-ranking π until (almost) all ties are broken. We say "almost" because this scheme cannot break a tie between distinct u, v if $u \equiv v$. These ties are broken arbitrarily as the final step of the algorithm (equivalently, the result is refined using an arbitrary full-ranking). If all of $\pi_1, \ldots, \pi_k$ are full-rankings (i.e. we are given input to RANKAGG), then REPEATCHOICE is, in fact, equivalent to algorithm *pick-a-perm*. It is also possible to view REPEATCHOICE as a radix sort with random order of the digits.

Theorem 1 REPEATCHOICE *is a randomized expected* 2*-approximation algorithm for* PARTRANKAGG *with polynomial running time. The algorithm can be derandomized.*

[5] In the preliminary version [1] the term *shattering* was used for the same operator.

Proof The number of iterations is at most k and the running time is therefore clearly polynomial. We show the approximation guarantee. Fix two distinct $u, v \in V$. If $u \not\equiv v$, then it is obvious that $u <_\pi v$ with probability $w_{uv}/(w_{uv} + w_{vu})$ and $v <_\pi u$ with the remaining probability $w_{vu}/(w_{uv} + w_{vu})$. Hence, the total expected cost of the algorithm is

$$\begin{aligned}\mathbf{E}[\text{cost}(\pi)] &= \sum_{u \not\equiv v} \left(\frac{w_{uv}}{w_{uv} + w_{vu}} w_{vu} + \frac{w_{vu}}{w_{uv} + w_{vu}} w_{uv} \right) \\ &= \sum_{u \not\equiv v} \frac{2 w_{uv} w_{vu}}{w_{uv} + w_{vu}} \leq 2 \sum_{u \not\equiv v} \min\{w_{uv}, w_{vu}\}. \end{aligned} \tag{1}$$

(Both summations are over *unordered* distinct pairs u, v.) On the other hand, any optimal solution π^* satisfies $\text{cost}(\pi^*) \geq \sum_{u \not\equiv v} \min\{w_{uv}, w_{vu}\}$. Hence, $\mathbf{E}[\text{cost}(\pi)] \leq 2\text{cost}(\pi^*)$, as required.

We now show how to derandomize the choice of π_i for iteration t. Assume we chose $\pi_{i_1}, \ldots, \pi_{i_{t-1}}$ for distinct $i_1, \ldots, i_{t-1} \in [k]$ in the previous steps. Let $\pi^{(t-1)}$ denote the intermediate partial ranking computed after the $(t-1)$'th iteration of the main loop of RepeatChoice, namely, $\pi^{(t-1)} = \pi_{i_{t-1}} * \cdots * \pi_{i_1} * \mathbf{1}_V$. Splitting the expected cost into past, present and future, we have:

$$\begin{aligned}\mathbf{E}[\text{cost}(\pi) \mid i_1, \ldots, i_t] = \frac{1}{k} \sum_{j=1}^{k} d(\pi^{(t-1)}, \pi_j) \\ + \sum_{(u,v) \in P_<} w_{vu} \\ + \sum_{\{u,v\} \in P_=} 2 \frac{w_{uv} w_{vu}}{w_{uv} + w_{vu}},\end{aligned}$$

where $P_<$ denotes all ordered pairs (u, v) such that $u =_{\pi^{(t-1)}} v$ and $u <_{\pi_t} v$, and $P_=$ denotes all unordered nonequivalent pairs $\{u, v\}$ such that $u =_{\pi^{(t-1)}} v$ and $u =_{\pi_t} v$. Computing $\mathbf{E}[\text{cost}(\pi)|i_1, \ldots, i_t]$ can be clearly done efficiently, and choosing i_t minimizing it at each step guarantees (by the principle of conditional expectations) a deterministic 2-approximation, as required. □

4 A (3/2)-Approximation Algorithm for PartRankAgg

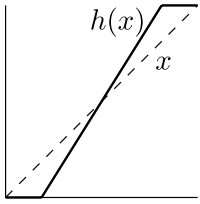
The 2-approximation algorithm described in Sect. 3 took advantage of the fact that any optimal solution had to pay the minimum of w_{uv} and w_{vu} for any pair $u, v \in V$. In this section we take advantage of additional structure arising from considering triplets $u, v, y \in V$.

Fix three distinct elements $u, v, y \in V$. Clearly, $w_{uv} \leq w_{uy} + w_{yv}$, because any vote π_i that ranked u strictly before v must have either ranked u strictly before y or y strictly before v. This inequality is known as the triangle inequality on the

weights induced by $\pi_1, \ldots, \pi_k$. In [3, 4], a (4/3)-approximation algorithm is presented for RANKAGG. The (rather complicated) analysis relies heavily on the fact that the input is a list of full-rankings (and not just any partial-ranking). The algorithm there involves (1) solving an LP relaxation for RANKAGG and using a randomized variant of the Quicksort algorithm for rounding it, (2) running pick-a-perm, and (3) outputting the better of the two results. In this section we consider the same LP and present a more complicated rounding technique, called LPKWIKSORT$_h$. This will result in the (3/2)-approximation algorithm for PARTRANKAGG. Our scheme does not involve taking the best of two algorithms, though it is quite possible that the best of LPKWIKSORT$_h$ and REPEATCHOICE gives a (4/3)-algorithm for PARTRANKAGG (we leave this as a question for future work). Our new rounding technique LPKWIKSORT$_h$ is, as far as we know, the first algorithm that beats the 2-approximation for RANKAGG by itself, without the need of running pick-a-perm and taking the best of the two. It is also the first algorithm that beats the 2-approximation for general minimum feedback arc-set in weighted tournaments with the triangle inequality [3].

To describe our improved algorithm, we first define a piecewise-linear function h mapping the real interval $[0, 1]$ onto itself. The function is defined as follows (see Sect. 7 for notes on the function):

$$h(x) = \begin{cases} 0 & \text{if } 0 \le x \le \frac{1}{6}, \\ \frac{3}{2}x - \frac{1}{4} & \text{if } \frac{1}{6} < x \le \frac{5}{6}, \\ 1 & \text{if } \frac{5}{6} < x \le 1. \end{cases}$$



Note that h is symmetric in the sense that for all $x \in [0, 1]$, $h(x) = 1 - h(1 - x)$ (in particular $h(1/2) = 1/2$). The function h will be used in a rounding algorithm of the following standard LP relaxation of PARTRANKAGG. The LP has a variable x_{uv} for each ordered $u \neq v$:

$$\begin{aligned} \text{minimize} \quad & \sum_{u \neq v} (x_{uv} w_{vu} + x_{vu} w_{uv}) \\ \text{s.t.} \quad & x_{uv} + x_{vu} = 1 \quad \forall u, v, \\ & x_{uv} \le x_{uy} + x_{yv} \quad \forall u, v, y, \\ & x_{uv} \ge 0 \quad \forall u, v. \end{aligned} \tag{2}$$

Clearly, if we could enforce $x_{uv} \in \{0, 1\}$ for all u, v, we would have an exact IP for PARTRANKAGG. Hence the LP is a valid relaxation for PARTRANKAGG. Given an optimal fractional solution $\{x_{uv}\}_{u,v}$, we round it using LPKWIKSORT$_h$ (Fig. 2), which returns a list of all elements in V in some order. We convert this list into a full-ranking (e.g. a mapping onto $[n]$) in the obvious way (the first maps to 1, the second to 2, and so on). The algorithm improves the LP rounding algorithm in [3], by using h to bias probabilities of placing vertices on either side of the pivot vertex. If we used the rounding algorithm there without biasing first, our proof techniques would not give an approximation factor of better than 2 (see Sect. 7 and [3]).

```
LpKwikSorth (V, x = {x_uv}_{u,v∈V})

if V = ∅ then
  return empty list
set L ← ∅, R ← ∅
pick pivot v ∈ V uniformly at random
for all u ∈ V, u ≠ v
  with probability h(x_uv)
    add u to L
  else (with remaining probability 1 − h(x_uv) = h(x_vu))
      add u to R
  return concatenation of:
    LpKwikSort_h (L, x), v, LpKwikSort_h (R, x)
```

Fig. 2 Pseudocode for LPKWIKSORT$_h$

Theorem 2 LPKWIKSORT$_h$ *returns a full-ranking* π *with an expected cost of at most* $3/2$ *times the optimal LP value* (*and hence also the optimal cost of* PARTRANKAGG).

Note that this implies a bound of $3/2$ on the LP integrality gap. The proof technique is very similar to [3, 4]. The main difficulty here is the use of the h-function and applicability to partial-rankings.

Proof The basic idea is to decompose the costs into so-called *backward* costs (corresponding to triplets in V) and *forward* costs (corresponding to pairs in V). For each $t = \{u, v, y\} \in T$, we define an event A_t in the random space induced by the execution of LPKWIKSORT$_h$. To define this event, we first notice that all elements of V are chosen as the pivot at some point in the recursive execution of LPKWIKSORT$_h$. We say that A_t occurred if when the first among u, v, y is chosen as pivot, the other two were input to the same recursive call. Note that conditioned on A_t, all three among u, v, y are equally likely to be that pivot (because the pivot is chosen uniformly at random). Let p_t denote $\Pr[A_t]$. Assume A_t occurs (and, say, v is the pivot). In that case, we will charge a *backward* cost to t corresponding to the random placement (i.e. L (left) or R (right)) of u, y. If u is placed in L and y in R, then the charge is w_{yu}, and if y is placed in L and u in R then the charge is w_{uy} (in all other cases, the charge is 0). It is immediate to verify that the total expected backward cost is $B = \sum_{t \in T} \frac{1}{3} p_t f(t)$, where for $t = \{u, v, y\}$,

$$\begin{aligned} f(t) = {} & h(x_{uv})h(x_{yu})w_{vy} + h(x_{vu})h(x_{uy})w_{yv} \\ & + h(x_{vy})h(x_{uv})w_{yu} + h(x_{yv})h(x_{vu})w_{uy} \\ & + h(x_{yu})h(x_{vy})w_{uv} + h(x_{uy})h(x_{yv})w_{vu}. \end{aligned} \tag{3}$$

For $e = \{u, v\} \in E$, let C_e denote the event that when the first among u, v was chosen as pivot, the other was input to the same recursive call. Let q_e denote $\Pr[C_e]$. Conditioned on C_e, we assign a *forward* cost to e, defined as w_{vu} if u is on the

left of v and w_{uv} otherwise. If C_e doesn't occur, the corresponding forward cost is 0. The expected forward cost charged to e conditioned on C_e is hence $\hat{c}(e) = h(x_{uv})w_{vu} + h(x_{vu})w_{uv}$, and the total expected forward cost is $F = \sum_{e\in E} q_e \hat{c}(e)$. The expected cost of the ranking returned by LpKwikSort$_h$ is $B + F$.

The LP value is $\sum_{e\in E} c(e)$, where $c(e) = (x_{uv}w_{vu} + x_{vu}w_{uv})$ for $e = \{u, v\}$. We decompose this as $B_{LP} + F_{LP}$, where $B_{LP} = \sum_{t\in T} \frac{1}{3} p_t g(t)$, for all $t = \{u, v, y\} \in T$:

$$\begin{aligned} g(t) = {} & (h(x_{uv})h(x_{yu}) + h(x_{vu})h(x_{uy}))c(\{vy\}) \\ & + (h(x_{vy})h(x_{uv}) + h(x_{yv})h(x_{vu}))c(\{yu\}) \\ & + (h(x_{yu})h(x_{vy}) + h(x_{uy})h(x_{yv}))c(\{uv\}), \end{aligned} \tag{4}$$

and $F_{LP} = \sum_{e\in E} q_e c(e)$. We claim that $\sum_{e\in E} c(e) = B_{LP} + F_{LP}$. To see this, we need to verify that the sum of the coefficients of $c(e)$ on the RHS is exactly 1 for all $e \in E$. Indeed, fix $e = \{u, v\} \in E$. The variable $c(e)$ appears in B_{LP} with total coefficient

$$\text{coeff}_{B_{LP}}[c(e)] = \sum_{y\in V\setminus\{u,v\}} \frac{1}{3} p_{\{u,v,y\}}(h(x_{uy})h(x_{yv}) + h(x_{yu})h(x_{uy})).$$

Each summand in the last expression is the probability of an event we denote by $D_{\{u,v\},y}$ defined as follows: at some point all of u, v, y are in the input to the same recursive call of LpKwikSort$_h$, y is chosen as pivot and separates between u and v (the 1/3 comes from the fact that conditioned on one of u, v, y being chosen as pivot, the probability of the choice being y is 1/3). The coefficient in F_{LP} is $\text{coeff}_{F_{LP}}[c(e)] = q_e = \Pr[C_e]$. The collection $\{D_{\{u,v\},y} : y \in V \setminus \{u, v\}\} \cup \{C_e\}$ is a disjoint cover of the probability space, because the order of u and v is determined exactly once in the execution of LpKwikSort$_h$. Hence, the sum of their probabilities is 1.

We want to show that $(B + F)/(B_{LP} + F_{LP}) \le 3/2$ whenever $B_{LP} + F_{LP} > 0$ and that if $B_{LP} + F_{LP} = 0$ then $B + F = 0$. To do so, it suffices to show that (i) for all $e \in E$, $\hat{c}(e)/c(e) \le 3/2$ whenever $c(e) > 0$ and $\hat{c}(e) = 0$ otherwise, and, (ii) for all $t \in T$, $f(t)/g(t) \le 3/2$ whenever $g(t) > 0$ and $f(t) = 0$ otherwise. Proof of (ii) is deferred to Lemma 1.

To prove (i), it suffices to show (slightly changing notation) that for all $x, w_1, w_2 \in [0, 1]$ such that $w_1 + w_2 \le 1$, $(h(x)w_1 + (1 - h(x))w_2)/(xw_1 + (1 - x)w_2) \le 3/2$ whenever the denominator is positive, and that the numerator is 0 if the denominator is 0. Define $a(x, w_1, w_2) = (h(x)w_1 + (1 - h(x))w_2)/(xw_1 + (1 - x)w_2)$. If $w_1 = w_2 = 0$ then both the numerator and the denominator are 0, so assume $w_1 + w_2 > 0$. Since we can divide w_1 and w_2 by $w_1 + w_2$ without changing the value of $a(x, w_1, w_2)$, we can assume $w_1 + w_2 = 1$. The function $b(x, w_1) = a(x, w_1, 1 - w_1)$ is a ratio of two linear functions in w_1 (for fixed x) and hence achieves its optima at $w_1 = 0$ or $w_1 = 1$. In the former case, $b(x, 0) = (1 - h(x))/(1 - x)$ and in the latter $b(x, 1) = h(x)/x$. By symmetry of h, it suffices to prove that $h(x)/x \le 3/2$ whenever $x \ne 0$, but clearly the maximum of $h(x)/x$ is obtained at $x = 5/6$, where $h(5/6)/(5/6) = 6/5 < 3/2$, as required. □

Lemma 1 *Fix $t = \{u, v, y\} \in T$. Consider the restriction*

$$\mathbf{w} = (w_{uv}, w_{vu}, w_{vy}, w_{yv}, w_{yu}, w_{uy}) \in \mathbf{R}^6$$

of the pairwise weight vector and the restriction

$$\mathbf{x} = (x_{uv}, x_{vu}, x_{vy}, x_{yv}, x_{yu}, x_{uy}) \in \mathbf{R}^6$$

of the LP solution to pairs in t. Define $f(t), g(t)$ as in (3) and (4). Then $f(t)/g(t) \le 3/2$ if $g(t) > 0$ and $f(t) = 0$ otherwise.

Proof Denote by Δ_w the space of all possible vectors $\mathbf{w}$. By the definition of pairwise weights, Δ_w is the convex closure of points $\mathbf{w}$ corresponding to *individual* votes on u, v, y. Indeed, there are 13 such votes: the 6 permutations $[u, v, y]$, $[u, y, v]$, $[v, u, y]$, $[v, y, u]$, $[y, u, v]$, $[y, v, u]$, the 6 votes tying 2 elements $[u, vy]$, $[v, yu]$, $[y, uv]$, $[uv, y]$, $[vy, u]$, $[yu, v]$ and the all-tie vote $[uvy]$. The corresponding points of Δ_w are (for example) $\mathbf{w} = (1, 0, 1, 0, 0, 1)$ (corresponding to $[u, v, y]$), $\mathbf{w} = (1, 0, 0, 0, 0, 1)$ (corresponding to $[u, vy]$) and $\mathbf{w} = (0, 0, 0, 0, 0, 0)$ (corresponding to $[uvy]$).

To simplify the proof, instead of working with the ratio $f(t)/g(t)$, we can equivalently show that $z(t) := f(t) - \frac{3}{2}g(t) \le 0$. The function $z(t)$ is linear in w_{uv}, w_{vu}, w_{vy}, w_{yv}, w_{yu} and w_{uy} (assuming fixed $\mathbf{x}$) and hence achieves its optima on vertices of Δ_w. Due to symmetry, and without loss of generality, we can consider 3 vertices. The first one is $\mathbf{w} = (0, 0, 0, 0, 0, 0)$, for which clearly $z(t) \equiv 0$.

The other two vertices are a permutation (say, $\mathbf{w} = (1, 0, 1, 0, 0, 1)$) and a 2-tie vote (say, $\mathbf{w} = (1, 0, 0, 0, 0, 1)$. We therefore consider two functions, corresponding to the two substitutions for $\mathbf{w}$. Substituting the two $\mathbf{w}$'s into z, the two corresponding functions $r(\mathbf{x})$ and $s(\mathbf{x})$ are

$$\begin{aligned} r(\mathbf{x}) &= \big(h(x_{uv})h(x_{yu}) + h(x_{yv})h(x_{vu}) + h(x_{yu})h(x_{vy})\big) \\ &\quad - \frac{3}{2}\big(\big(h(x_{uv})h(x_{yu}) + h(x_{vu})h(x_{uy})\big)x_{yv} \\ &\quad + \big(h(x_{vy})h(x_{uv}) + h(x_{yv})h(x_{vu})\big)x_{yu} \\ &\quad + \big(h(x_{yu})h(x_{vy}) + h(x_{uy})h(x_{yv})\big)x_{vu}\big), \qquad (5) \\ s(\mathbf{x}) &= \big(h(x_{yv})h(x_{vu}) + h(x_{yu})h(x_{vy})\big) \\ &\quad - \frac{3}{2}\big(\big(h(x_{vy})h(x_{uv}) + h(x_{yv})h(x_{vu})\big)x_{yu} \\ &\quad + \big(h(x_{yu})h(x_{vy}) + h(x_{uy})h(x_{yv})\big)x_{vu}\big). \end{aligned}$$

We may substitute $1 - x_{uv}$ for x_{vu}, $1 - x_{vy}$ for x_{yv} and $1 - x_{yu}$ for x_{uy} (from the LP constraints). Abusing notation, we keep using s, r to denote the functions after the substitution, and $\mathbf{x} \in \mathbf{R}^3$ to denote (x_{uv}, x_{vy}, x_{yu}). By the triangle inequality, we have $1 \le x_{uv} + x_{vy} + x_{yu} \le 2$. Let $\Delta_x \subseteq \mathbf{R}^3$ denote the polytope of possible $\mathbf{x}$'s. The function h is linear on each one of the intervals $I_1 = [0, 1/6]$, $I_2 = [1/6, 5/6]$ and $I_3 = [5/6, 1]$, and hence it will be useful to separately analyze s and r on each of

Table 1 Analysis of s on 27 domains. The entries $(i, j, k) = (1, 1, 1)$ and $(i, j, k) = (3, 3, 3)$ do not appear as $D_{i,j,k} = \emptyset$ there

i, j, k	$s\|_{D_{i,j,k}}$	max s	argmax s
1, 1, 2	$(-7 + 6x_3 - 3x_1(-5 + 6x_3))/8$	$-1/4$	$(\frac{5}{64}, \frac{1}{8}, \frac{5}{6})$
1, 1, 3	$1 - 3x_3/2$	$-1/4$	$(\frac{1}{6}, 0, \frac{5}{6})$
1, 2, 1	$-(-5 + 6x_2)(-1 + 3x_1 - 3x_3)/8$	0	$(\frac{5}{64}, \frac{5}{6}, \frac{1}{8})$
1, 2, 2	$3(-6 + 8x_2 + 6x_3 - 12x_2x_3 + x_1(13 - 18x_3 + 18x_2(-1 + 2x_3)))/16$	0	$(\frac{5}{64}, \frac{5}{6}, \frac{1}{6})$
1, 2, 3	$(11 + 3x_1(-1 + 6x_2) + 18x_2(-1 + x_3) - 15x_3)/8$	$-1/4$	$(\frac{1}{6}, \frac{1}{4}, \frac{5}{6})$
1, 3, 1	0	0	Const. func.
1, 3, 2	$(-1 + 3x_1)(-1 + 6x_3)/8$	0	$(\frac{5}{64}, \frac{59}{64}, \frac{1}{6})$
1, 3, 3	$(-1 + 3x_1)/2$	$-1/4$	$(\frac{1}{6}, \frac{5}{6}, \frac{5}{6})$
2, 1, 1	$(-2 + 3(-5 + 6x_1)x_3)/8$	$-1/4$	$(\frac{5}{6}, \frac{1}{8}, \frac{5}{64})$
2, 1, 2	$(-5 + 3x_1 + 3x_3)/8$	0	$(\frac{5}{6}, \frac{5}{64}, \frac{5}{6})$
2, 1, 3	$(-5 + 6x_1)(-2 + 3x_3)/8$	0	$(\frac{5}{6}, \frac{5}{64}, \frac{59}{64})$
2, 2, 1	$(-5 + (-39 + 54x_1)x_3 + x_2(6 - 54(-1 + 2x_1)x_3))/16$	0	$(\frac{1}{6}, \frac{5}{6}, \frac{3}{128})$
2, 2, 2	$(-13 + x_1(9 - 18x_2) - 18x_2(-1 + x_3) + 9x_3)/16$	0	$(\frac{5}{6}, \frac{1}{6}, \frac{5}{6})$
2, 2, 3	$-3(-9 + 13x_3 + 6x_1(-1 + 2x_2)(-2 + 3x_3) - 2x_2(-7 + x_3))/16$	0	$(\frac{5}{6}, \frac{1}{6}, \frac{59}{64})$
2, 3, 1	$-3(-1 + 6x_1)x_3/8$	0	$(\frac{1}{6}, \frac{59}{64}, \frac{5}{64})$
2, 3, 2	$(1 - 3x_1 - 3x_3)/8$	0	$(\frac{1}{6}, \frac{59}{64}, \frac{1}{6})$
2, 3, 3	$(-4 + 3x_3 - 6x_1(-2 + 3x_3))/8$	$-1/4$	$(\frac{1}{6}, \frac{7}{8}, \frac{59}{64})$
3, 1, 1	$3(-1 + x_1)/2$	0	$(1, 0, 0)$
3, 1, 2	$-3(-1 + x_1)(-5 + 6x_3)/8$	0	$(1, \frac{5}{64}, \frac{1}{2})$
3, 1, 3	0	0	Const. func.
3, 2, 1	$-3(5 + x_1(-5 + 6x_2) + 6x_2(-1 + x_3) - x_3)/8$	0	$(1, \frac{9}{32}, 0)$
3, 2, 2	$(-38 + 54x_3 - 12x_2(-4 + 9x_3) + 3x_1(13 - 18x_3 + 18x_2(-1 + 2x_3)))/16$	0	$(1, \frac{1}{6}, \frac{1}{2})$
3, 2, 3	$(-1 + 6x_2)(-1 + 3x_1 - 3x_3)/8$	0	$(\frac{59}{64}, \frac{1}{6}, \frac{7}{8})$
3, 3, 1	$-3x_3/2$	0	$(\frac{5}{6}, \frac{5}{6}, 0)$
3, 3, 2	$(1 - 18x_3 + 3x_1(-1 + 6x_3))/8$	$-1/4$	$(\frac{59}{64}, \frac{7}{8}, \frac{1}{6})$

the 27 domains $D_{i,j,k} = \Delta_x \cap (I_i \times I_j \times I_k)$ for $i, j, k \in \{1, 2, 3\}$. The remainder of the proof is elementary case-by case analysis of the multinomials s and r on the 27 corresponding domains. In Tables 1 and 2 we present the functions s and r (respectively) after substitution on these domains. For ease of notation in the table, we use x_1, x_2 and x_3 instead of x_{uv}, x_{vy} and x_{yu}, respectively. To simplify the verification of the tables, note that the restriction of s and r to $D_{i,j,k}$ for all i, j, k is *trilinear* (linear in each of the three variables x_1, x_2, x_3 when the other two are fixed). Therefore,

Table 2 Analysis of r on 27 domains. The entries $(i, j, k) = (1, 1, 1)$ and $(i, j, k) = (3, 3, 3)$ do not appear as $D_{i,j,k} = \emptyset$ there

i, j, k	$r\|_{D_{i,j,k}}$	$\max r$	$\operatorname{argmax} r$
1, 1, 2	$(-22 + 24x_3 - 3x_1(-5 + 6x_3) - 3x_2(-5 + 6x_3))/8$	$-1/4$	$(\frac{5}{64}, \frac{1}{8}, \frac{5}{6})$
1, 1, 3	$1 - 3x_3/2$	$-1/4$	$(\frac{1}{6}, 0, \frac{5}{6})$
1, 2, 1	$(-17 - 3x_1(-5 + 6x_2) - 15x_3 + 18x_2(1 + x_3))/8$	$-1/4$	$(\frac{5}{64}, \frac{5}{6}, \frac{1}{8})$
1, 2, 2	$3(2(-8 + x_2(9 - 12x_3) + 9x_3)$ $+ x_1(13 - 18x_3 + 18x_2(-1 + 2x_3)))/16$	$-1/4$	$(\frac{1}{6}, \frac{11}{64}, \frac{5}{6})$
1, 2, 3	$(11 + 3x_1(-1 + 6x_2) + 18x_2(-1 + x_3) - 15x_3)/8$	$-1/4$	$(\frac{1}{6}, \frac{1}{4}, \frac{5}{6})$
1, 3, 1	$3(-1 + x_2)/2$	0	$(0, 1, 0)$
1, 3, 2	$(-14 + 15x_2 + 12x_3 - 18x_2x_3 + 3x_1(-1 + 6x_3))/8$	0	$(\frac{5}{64}, 1, \frac{1}{6})$
1, 3, 3	$(-1 + 3x_1)/2$	$-1/4$	$(\frac{1}{6}, \frac{5}{6}, \frac{5}{6})$
2, 1, 1	$(-17 + 15x_2 - 18x_1(-1 + x_2 - x_3) - 15x_3)/8$	$-1/4$	$(\frac{5}{6}, \frac{5}{64}, \frac{1}{8})$
2, 1, 2	$3(-16 + 13x_2 + 18x_3 - 18x_2x_3$ $+ 6x_1(3 - 4x_3 + x_2(-3 + 6x_3)))/16$	$-1/4$	$(\frac{5}{6}, \frac{5}{64}, \frac{1}{6})$
2, 1, 3	$(11 - 3x_2 - 15x_3 + 18x_1(-1 + x_2 + x_3))/8$	$-1/4$	$(\frac{1}{6}, \frac{5}{64}, \frac{5}{6})$
2, 2, 1	$(-35 - 39x_3 + 18x_2(2 + 3x_3)$ $-18x_1(-2 - 3x_3 + x_2(2 + 6x_3)))/16$	0	$(\frac{5}{6}, \frac{5}{6}, 0)$
2, 2, 2	$3(-17 + x_2(19 - 24x_3) + 19x_3$ $+ x_1(19 - 24x_3 + 12x_2(-2 + 3x_3)))/16$	$-7/128$	$(\frac{5}{6}, \frac{7}{12}, \frac{7}{12})$
2, 2, 3	$(29 - 6x_1(8 + 18x_2(-1 + x_3) - 9x_3)$ $- 39x_3 + 6x_2(-8 + 9x_3))/16$	$-7/128$	$(\frac{7}{12}, \frac{7}{12}, \frac{5}{6})$
2, 3, 1	$-3(5 - 5x_2 - x_3 + 6x_1(-1 + x_2 + x_3))/8$	0	$(\frac{5}{6}, \frac{57}{64}, 0)$
2, 3, 2	$3(-12 + 13x_2 + 14x_3 - 18x_2x_3$ $+ 2x_1(7 - 12x_3 + 9x_2(-1 + 2x_3)))/16$	0	$(\frac{1}{6}, 1, \frac{1}{6})$
2, 3, 3	$3(-1 - x_2 + x_1(2 + 6x_2 - 6x_3) + x_3)/8$	$-1/8$	$(\frac{1}{3}, \frac{5}{6}, \frac{5}{6})$
3, 1, 1	$3(-1 + x_1)/2$	0	$(1, 0, 0)$
3, 1, 2	$(-14 - 3x_2 + 12x_3 + 18x_2x_3 - 3x_1(-5 + 6x_3))/8$	0	$(1, \frac{5}{64}, \frac{1}{6})$
3, 1, 3	$(-1 + 3x_2)/2$	$-1/4$	$(\frac{5}{6}, \frac{1}{6}, \frac{5}{6})$
3, 2, 1	$-3(5 + x_1(-5 + 6x_2) + 6x_2(-1 + x_3) - x_3)/8$	0	$(1, \frac{9}{32}, 0)$
3, 2, 2	$3(2(-6 + x_2(7 - 12x_3) + 7x_3)$ $+ x_1(13 - 18x_3 + 18x_2(-1 + 2x_3)))/16$	0	$(1, \frac{1}{6}, \frac{1}{6})$
3, 2, 3	$3(-1 + x_1(-1 + 6x_2) + x_2(2 - 6x_3) + x_3)/8$	$-1/8$	$(\frac{5}{6}, \frac{1}{3}, \frac{5}{6})$
3, 3, 1	$-3x_3/2$	0	$(\frac{5}{6}, \frac{5}{6}, 0)$
3, 3, 2	$(2 - 24x_3 + 3x_1(-1 + 6x_3) + 3x_2(-1 + 6x_3))/8$	$-1/8$	$(\frac{5}{6}, \frac{5}{6}, \frac{1}{3})$

any face of the polytope $D_{i,j,k}$ parallel to an axis can be removed from consideration when studying the maxima of s, r on $D_{i,j,k}$ (because such faces are unions of axis-parallel line segments on which s, r are linear; the endpoints of these line segments

are contained in lower-dimensional faces).[6] It remains to consider only the 2, 1 and 0-dimensional faces of $D_{i,j,k}$ parallel to the hyperplanes defined by $x_1 + x_2 + x_3 = 2$ and $x_1 + x_2 + x_3 = 1$. □

5 2RATINGAGG is in P, TOP2AGG is NP-Hard

Theorem 3 *p*RATINGAGG *has a polynomial-time algorithm when* $p = 2$ (*hence, so does* TOP*m*AGG *for* $m = 1$).

Proof Given an instance $\pi_1, \ldots, \pi_k : V \to [2]$ of 2RATINGAGG, for each $v \in V$ we let n_v be the number of integers $i \in [k]$ such that $\pi_i(v) = 2$. It is easy to see that $n_u > n_v \Leftrightarrow w_{vu} > w_{uv}$. Therefore, if for some full-ranking $\pi : V \to [n]$ there are $u, v \in V$ such that $\pi(u) + 1 = \pi(v)$ and $n_u > n_v$, then the cost of π will strictly improve if we swap the values of π at u and v. Hence sorting V in increasing n_v order (breaking ties arbitrarily) are exactly the optimal rankings. Such a ranking can be computed in polynomial time. □

Theorem 4 TOP*m*AGG *is NP-Hard when* $m = 2$ (*hence, so is p*RATINGAGG *for* $p = 3$).

Proof We show a reduction from minimum feedback arc-set in tournaments (MINFASTOUR), which was recently shown to NP-Hard by Noga Alon [5] (based on a derandomization of a reduction from [3]). MINFASTOUR is the problem of, given a tournament $T = (V, A)$, finding a ranking π of V minimizing the number of backward edges, namely $\text{cost}_T(\pi) = \sum_{u <_\pi v} \mathbf{1}_{(v,u) \in A}$, where $\mathbf{1}_P$ is 1 if predicate P is true and 0 otherwise.

Given an instance $T = (V, A)$ of MINFASTOUR, we define a corresponding instance of TOP2AGG. The votes $\pi_{\{u,v\}} : V \to [3]$ are indexed using all $\binom{n}{2}$ unordered pairs $\{u, v\}$, where for each $\{u, v\}$ and $y \neq u, v$:

$$\pi_{\{u,v\}}(u) = \begin{cases} 1, & (u, v) \in A, \\ 2, & (v, u) \in A, \end{cases}$$

$$\pi_{\{u,v\}}(y) = 3.$$

It is not hard to see that for this instance, any full-ranking π of V has cost

$$\text{cost}(\pi) = \frac{1}{\binom{n}{2}} \sum_{u <_\pi v} (\mathbf{1}_{(v,u) \in A} + (n - 2)).$$

(The $n - 2$ term is from the contribution of $\pi_{\{v,y\}}$ for $y \neq u, v$.) Therefore, $\text{cost}(\pi)$ and $\text{cost}_T(\pi)$ are linearly related. This completes the NP-Hardness reduction. □

[6] Faces of polytopes are *open* sets (in their affine closure) by convention.

6 Applicability to Other Metrics on Partial-Rankings

As mentioned in the introduction, d belongs to a family of distance functions studied by Fagin et al. [15]. This family is parametrized by a real number $0 \leq \rho \leq 1$. Our distance function d is obtained by taking $\rho = 0$ and is not a proper distance function because the distance between two partial-rankings may be zero (in fact, even if one of the arguments is a full-ranking).[7] The distance $d_\rho(\sigma, \pi)$ between two partial-rankings σ, π is defined in [15] as $d(\sigma, \pi) + \rho d'(\sigma, \pi)$, where $d'(\sigma, \pi)$ is the number of unordered pairs $\{u, v\}$ such that either $u =_\pi v$ and $u \neq_\sigma v$, or $u =_\sigma v$ and $u \neq_\pi v$. The function $d'(\sigma, \pi)$ measures the difference between the two clusterings of V induced by the tie-relations in σ and π (in fact, it is exactly the *consensus-clustering* metric [18, 26, 27]). It is shown in [15] that $d_{1/2}$ is a metric belonging to an important class $\mathcal{D}$ of many other equivalent metrics derived from both the Kendall-τ and the Spearman's footrule metrics on full-rankings. We will not go into the definitions of the metrics belonging to $\mathcal{D}$ studied in [15].

If we used $d_{1/2}$ instead of d in the definition of our objective function, this would add a nonnegative constant depending on the input $\pi_1, \ldots, \pi_k$ (and not on the output). Indeed, since an output σ is a full-ranking, $d'(\sigma, \pi_i)$ is simply the number of pairs u, v that are tied in π_i. Therefore, by Theorems 1 and 2, algorithms RepeatChoice and LpKwikSort$_h$ are respectively 2 and 3/2 factor approximation algorithms with respect to $d_{1/2}$. By up-to-constant equivalence of all the metrics in $\mathcal{D}$, this also implies constant-factor approximations with respect to them as well.

Partial-rankings can be thought of as *partially revealed* full-rankings. Assuming this interpretation, the metrics in class $\mathcal{D}$ can be thought of as methods for compensating for unrevealed information. We suggest the following alternative approach for compensating. For a partial-ranking π, let $\mu(\pi)$ count the number of pairs u, v that are *not* tied in π (for full-rankings, this is $\binom{n}{2}$). Intuitively, μ measures the amount of information revealed by π. For a full-ranking σ and partial-ranking π, define $\hat{d}(\sigma, \pi)$ as $d(\sigma, \pi)\binom{n}{2}/\mu(\pi)$ (if $\mu(\pi) = 0$ then also $d(\sigma, \pi) = 0$ and we define $\hat{d}(\sigma, \pi) = 0$). Intuitively, this spreads the available distance information evenly across the missing information. In fact, for aggregation of 2-ratings, the measure $1 - \hat{d}(\sigma, \pi)/\binom{n}{2}$ is known as the AUC (Area Under the ROC Curve) of σ with respect to the 2-rating (equivalently, binary classification) π. We suggest using $\hat{d}$ as an alternative to using d and to other distance measures that are used and studied in the literature. In fact, using $\hat{d}$ is a special case of the following weighted version of PartRankAgg:

Definition 5 Weighted-PartRankAgg is the problem of, given a list $\pi_1, \ldots, \pi_k$ of partial-rankings of V (votes) and nonnegative weights $w_1, \ldots, w_k$, outputting a full-ranking π minimizing $\text{cost}(\pi) = \frac{1}{k}\sum_{i=1}^{k} w_i d(\pi, \pi_i)$.

Indeed, we can take $w_i = \binom{n}{2}/\mu(\pi)$. Algorithms RepeatChoice and LpKwikSort$_h$ generalize to the weighted version. To see this, notice that an integer weight of ω assigned to a voter can be simulated by considering ω unweighted

[7] In [15] they use the term "distance function" for what we call "proper distance function", and aside from the definition, they don't deal much with $\rho = 0$.

copies of the voter. Also note that the hardness statement in Theorem 4 applies to the normalized version as well, because the hard instances in the proof assign uniform weights to the voters.

7 Concluding Remarks

1. Taking $h(x) = x$ in LPKWIKSORT$_h$ renders it equivalent to the LP rounding algorithm used in [3]. The approximation factor obtained using our techniques would be at least 2 with this choice of h, as the following example shows. Indeed, take $x_{uv} = x_{vy} = 1$, $x_{uy} = 1 - \varepsilon$, $w_{uv} = w_{vy} = w_{uy} = 1$, $w_{vu} = w_{yv} = w_{yu} = 0$, where $\varepsilon > 0$ is some small constant. It is immediate to verify that for the triangle $t = \{u, v, y\}$ we get $f(t) = 2\varepsilon$ and $g(t) = \varepsilon$, hence $f(t)/g(t) = 2$. This last example intuitively explains why we need a tweaking function h.
2. The choice of h is optimal in the sense that using any other h-function with the same rounding algorithm and the same analysis technique cannot result in a better than a 3/2 approximation (though different analysis might lead to a better result). Indeed, for $\mathbf{w} = (1, 0, 1, 0, 0, 1)$ and $\mathbf{x} = (1/2, 1/2, 1, 0, 1/2, 1/2)$ we get $f(t)/g(t) = 3/2$, and one can show easily that h must satisfy $h(0) = 0, h(1/2) = 1/2, h(1) = 1$. In order to come up with h, we first found the unique symmetric function $\hat{h}$ with the property that $f(t)/g(t) = 3/2$ for $\mathbf{w} = (1, 0, 1, 0, 0, 1)$ and $\mathbf{x} = (1-\alpha, \alpha, 1, 0, \alpha, 1-\alpha)$, for all $0 < \alpha \leq 1/2$. This function is $\hat{h}(x) = 1 - \sqrt{1 - \frac{3}{2}x}$ for $x \leq 1/2$ and its symmetric completion for $x > 1/2$. Then h is a piecewise linear approximation of $\hat{h}$. Working with $\hat{h}$ seems more difficult for analysis though may prove to be good in practice. Note that in an earlier version [1] a different, slightly more complicated h was used, but it was subsequently discovered that the one used here suffices.

8 Open Questions

1. Does there exist a PTAS for PARTRANKAGG? Does one exist for TOPmAGG or pRATINGAGG? Note that a PTAS has been recently found by Kenyon-Mathieu and Schudy [24] for RANKAGG.
2. Is taking the best of LPKWIKSORT$_h$ and REPEATCHOICE a 4/3-approximation for PARTRANKAGG (as the results in [3] would suggest)?
3. Does Coppersmith et al.'s [10] recent important result relating the Borda score-based rank aggregation algorithm with the Kemeny-optimal (for full-rankings) generalize to partial rankings?

Acknowledgements We would like to thank Moses Charikar, Alantha Newman, Ron Fagin, Cynthia Dwork, Kunal Talwar, Steve Chien, Seffi Naor, Joel Predd and Amit Agarwal for enlightening discussions and feedback, as well as anonymous reviewers for helpful comments, and in particular for a brilliant observation closing one of the open problems from previous versions.

References

1. Ailon, N.: Aggregation of partial rankings, p-ratings and top-m lists. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 415–424 (2007)
2. Ailon, N., Charikar, M.: Fitting tree metrics: Hierarchical clustering and phylogeny. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 73–82 (2005)
3. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC), pp. 684–693 (2005)
4. Ailon, N., Charikar, M., Newman, A.: Proofs of conjectures in 'Aggregating inconsistent information: Ranking and clustering'. Technical Report, Princeton University, TR-719-05 (2005)
5. Alon, N.: Ranking tournaments. SIAM J. Discrete Math. **20**(1), 137–142 (2006)
6. Arrow, K.J.: Social Choice and Individual Values. Wiley, New York (1951)
7. Aslam, J., Montague, M.: Condorcet fusion for improved retrieval. In: Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 538–548 (2002)
8. Borda, J.C.: Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences (1781)
9. Condorcet, M.-J.: Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix (1785)
10. Coppersmith, D., Fleischer, L., Rudra, A.: Ordering by weighted number of wins gives a good ranking for weighted tournamnets. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 776–782 (2006)
11. Diaconis, P., Graham, R.: Spearman's footrule as a measure of disarray. J. R. Stat. Soc. Ser. B **39**(2), 262–268 (1977)
12. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the Tenth International Conference on the World Wide Web (WWW10), pp. 613–622, Hong Kong (2001)
13. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation revisited. Manuscript (2001)
14. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E.: Comparing and aggregating rankings with ties. In: Proceedings of the ACM Symposium on Principles of Database Systems (PODS), pp. 47–58 (2004)
15. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E.: Comparing partial rankings. SIAM J. Discrete Math. **20**(3), 628–648 (2006)
16. Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 28–36, Baltimore (2003)
17. Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 301–312, San Diego (2003)
18. Filkov, V., Skiena, S.: Integrating microarray data by consensus clustering. In: Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI), pp. 418–425, Sacramento (2003)
19. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. TKDD **1**(1) (2007)
20. Hodge, J., Klima, R.E.: The Mathematics of Voting and Elections: A Hands-On Approach. Mathematical World, vol. 22. Am. Math. Soc., Providence (2000)
21. Kemeny, J., Snell, J.: Mathematical Models in the Social Sciences. Blaisdell, Boston (1962). Reprinted by MIT Press, Cambridge (1972)
22. Kemeny, J.G.: Mathematics without numbers. Daedalus **88**, 571–591 (1959)
23. Kendall, M., Gibbons, J.D.: Rank Correlation Methods. Arnold, Sevenoaks (1990)
24. Kenyon-Mathieu, C., Schudy, W.: How to rank with few errors. In: STOC'07: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, pp. 95–103. Assoc. Comput. Mach., New York, 2007
25. Rand, W.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
26. Strehl, A.: Relationship-based clustering and cluster ensembles for high-dimensional data mining. Ph.D. Dissertation, University of Texas at Austin, May 2002
27. Wakabayashi, Y.: The complexity of computing medians of relations. Resenhas **3**(3), 323–349 (1998)