Rank Aggregation Revisited

Cynthia Dwork[*] Ravi Kumar[†] Moni Naor[‡] D. Sivakumar[§]

Abstract

The *rank aggregation* problem is to combine many different rank orderings on the same set of candidates, or alternatives, in order to obtain a "better" ordering. Rank aggregation has been studied extensively in the context of social choice theory, where several "voting paradoxes" have been discovered. The problem also arises in many other settings:

> Sports and Competition: How to determine the winner of a season, how to rank players or how to compare players from different eras?
>
> Machine Learning: Collaborative filtering and meta-search;
>
> Statistics: Notions of Correlation;
>
> Database Middleware: Combining results from multiple databases.

A natural step toward aggregation was taken by Kemeny. Informally, given orderings $\tau_1, \ldots, \tau_k$ on (partial lists of) alternatives $\{1, 2, \ldots, n\}$, a *Kemeny optimal* ordering σ minimizes the sum of the "bubble sort" distances

$$\Sigma_{i=1}^{k} K(\sigma, \tau_i) .$$

Thus, intuitively, Kemeny optimal solutions produce "best" compromise orderings. However, finding a Kemeny optimal aggregation is NP-hard [4].

In this work we revisit rank aggregation with an eye toward reducing search engine spam in meta-search. We note the virtues of Kemeny optimal aggregation in this context, strengthen the NP-hardness results, and, most importantly, develop a natural relaxation called *local Kemeny optimality* that preserves the spam-fighting capabilities of Kemeny optimality at vastly reduced cost. We show how to efficiently take *any* initial aggregated ordering and produce a maximally consistent locally Kemeny optimal solution.

We therefore propose a new approach to rank aggregation: begin with any desirable initial aggregation and then "locally Kemenize" it. We also propose the use of Markov chains for obtaining the initial aggregation, and suggest four specific chains for this purpose.

[*]Compaq Systems Research Center, 130 Lytton Ave., Palo Alto, CA 94301. dwork@pa.dec.com

[†]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. ravi@almaden.ibm.com

[‡]Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Part of this work was done while the author was visiting the IBM Almaden Research Center. naor@wisdom.weizmann.ac.il

[§]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120. siva@almaden.ibm.com

1 Introduction

Informally, the *rank aggregation* problem is to combine many different rank orderings on the same set of candidates, or alternatives, in order to obtain a “better” ordering. Rank aggregation has been studied in many disciplines, most extensively in the context of social choice theory, where there is a rich literature dating from the latter half of the eighteenth century. We revisit rank aggregation with an eye toward meta-search. Specifically, our motivation is to combine results of several rank functions in order to strengthen resistance to search engine “spam,” or strategic manipulation of web pages in order to achieve an “undeservedly” high ranking.

There are many ways to aggregate rankings – one analysis of 27 democracies found that between 1945 and 1990 seventy differenct voting systems were in use for national legistlative elections [29, 13]. In 1770 Jean-Charles de Borda proposed “election by order of merit”: for each voter’s announced (linear) preference order on the alternatives, a score of 0 is assigned to the least preferred alternative, 1 to the next-to-least preferred, and so forth; then the total score of each alternative is computed and the one with the highest score is declared the winner.

To illustrate the range of possibilities, we briefly mention two other aggregation techniques. In the 1850’s, Thomas Hare (England) and George Andrae (Denmark) proposed the system of “single transferable voting.” Variants on this basic system are used in Australia, Malta, the Republic of Ireland, and Northern Ireland, in certain local elections in New York City and Cambridge, MA., and in elections of representatives of Northern Ireland to the European parliament. In this procedure, any candidate that receives more than a certain number of first-place votes is elected. Any “surplus” votes are distributed to the other candidates in accordance with the second-choice preferences of the voters. This process of electing and redistributing surplus votes is repeated until either no more positions remain to be filled or there are no more votes to be redistrubted but open positions remain. In the latter case the candidate with the *least* number of votes is eliminated and her/his votes are redistributed. If the system is used to elect a single alternative it is sometimes called *altenative voting* or *majority preference*. In this case redistribution of excess votes is not performed, only redistribution of the votes for eliminated cadidates.

In “cumulative voting,” voters cast as many votes as there are available seats; they may allocate multiple votes to a single candidate (precise details of allowable allocations vary). Winning candidates are determined by simple plurality. Used to elect Illinois state legislators from 1870 to 1980, this system has also been used to resolve voting rights cases for city council elections, county commission elections, and board elections in more than fifty jurisdictions. In 1994 cumulative voting was imposed by a federal judge in a Maryland voting rights case. It has also been used in British school board elections and in electing corporate boards of directors.

In these examples (as in many others not listed here) voters may submit what we call partial lists: rather than a ranking of all the alternatives, a partial list gives a (possibly weighted) prefix of a ranking. The goal is typically not to obtain a complete ordering on the alternatives, but rather to obtain a prefix of such an ordering (fill the available seats).

What are the desirable properties of an aggregated ranking? In 1785 Marie J. A. N. Caritat, Marquis de Condorcet, proposed that if there is some alternative, now known as the *Condorcet alternative*, that defeats every other in pairwise simple majority voting, then that alternative should be ranked first. A natural extension, due to Truchon, mandates that if there is a partition (T, U) of the alternatives $\{1, \ldots, n\}$ such that for any $x \in T$ and $y \in U$ the majority prefers x to y, then x must be ranked above y [39]. Interestingly, neither simple plurality voting nor any of the methods listed above (Borda, single tranferable voting, cumulative voting) ensure the election of the Condorcet winner, should one exist. An excellent survey of other popular criteria appears in [13] (see also [9]). In 1951 Arrow proved the fundamental theorem that no voting system can simulataneously ensure five natural and desirable fairness properties. Our work focusses on the extended Condorcet criterion. We will show that not only can it be achieved efficiently, but it also has excellent "spam-fighting" properties when used in the context of meta-search.

Although the paradox of voting – that the majority preference graph in an election with at least 3 candidates and at least 3 voters may contain a cycle – was known to Condorcet, its significance was not understood before a series of essays by Black, begun in the 1940's [13, 6]. Black suggested that the Condorcet winner should be chosen, should one exist, and otherwise the Borda winner should be selected [6]. As we next describe, our work generalizes Black's suggestion in two ways: we obtain an aggregated ranking on all the alternatives (we don't just find a single winner) and we show how to constructively combine this with any initial aggregation (not just the initial ranking given by the Borda scores).

Suppose there is an underlying "correct" ordering of alternatives ρ, and each order $\tau_1, \ldots, \tau_k$ is obtained from ρ by swapping two alternatives with some probability $p < 1/2$. Thus, the τ's are "noisy" versions of ρ. A Kemeny optimal aggregation σ of $\tau_1, \ldots, \tau_k$ is one that is maximally likely to have produced the τ's (it need not be unique). This maximum likelihood interpretation is due to Young [41]. We give a more operational definition next.

K-distance. Let π and σ be two partial lists of $\{1, \ldots, n\}$. The *K-distance* of π and σ, denoted $K(\pi, \sigma)$, is the number of pairs $i, j \in \{1, \ldots, n\}$ such that $\pi(i) < \pi(j)$ but $\sigma(i) > \sigma(j)$. Note that if it is not the case that both i and j appear in both lists π and σ, then the pair (i, j) contributes nothing to the K-distance between the two lists. For any two partial lists $K(\pi, \sigma) = K(\sigma, \pi)$, and if we restrict π and σ to be full lists (permutations), then K is a metric[1]. In this case K is known as the *Kendall-tau* distance between them and it corresponds to the number of transpositions *bubble sort* requires to turn π into σ.

SK, Kemeny optimal. For a collection of partial lists $\tau_1, \tau_2, \ldots, \tau_k$ and a full list π we denote

$$SK(\pi, \tau_1, \tau_2, \ldots, \tau_k) = \sum_{i=1}^{k} K(\pi, \tau_i)\,.$$

We say that a permutation σ is a *Kemeny optimal* aggregation of a collection of partial lists $\tau_1, \tau_2, \ldots, \tau_k$ if it minimizes $SK(\pi, \tau_1, \tau_2, \ldots, \tau_k)$ over all permutations π.

Kemeny optimal aggregations are of particular interest because they satisfy the extended Condorcet criterion. Let N be the set of alternatives and $\mathcal{P}(N)$ the set of partitions on elements in N. An equivalent reformulation of the extended Condorcet criterion says that if we consider any partition $\hat{N} \in \mathcal{P}(N)$, where for all $N_a, N_b \in \hat{N}$, where $a < b$, for all $x \in N_a$ and all $y \in N_b$, x is ranked above y by a majority of $\tau_1, \ldots, \tau_k$, then in the aggregation of the τ's x is ranked above y. If there are no cycles or ties in the majority relation, then all N_a of the finest partition of $\mathcal{P}(N)$ under consideration are singletons, and an aggregate ranking satisfying the extended Condorcet criterion is a complete order. If there are cycles or ties, the extended Condorcet criterion does not say how to order the alternatives within a partition. Kemeny optimal aggregations give a "best" ordering (under the K measure) within the partitions.

While it is not hard to compute $K(\pi, \sigma)$ in time $O(n \log n)$ (sophisticated data structures can improve the time to $O(n \log n / \log \log n)$ [2]), finding a Kemeny optimal aggregation is known to be NP-hard; we show this to be so even in the special case of 4 complete lists.

The principal contributions of this work are to provide an efficient approach to rank aggregation that guarantees satisfaction of the extended Condorcet criterion, and to establish the value of this criterion for meta-search. Our approach is to begin with *any* initial aggregation μ of $\tau_1, \ldots, \tau_k$ and then to efficiently compute a *"local Kemenization"* of μ – that is, to produce an odering σ that is, intuitively, as consistent with μ as possible while simultaneously satisfying the extended Condorcet criterion.

[1] This is not true in general, e.g., consider three lists one of which is empty; the distance to an empty list is always 0.

Thus, our approach generalizes Black's suggestion by (1) permitting any initial aggregation method, including but not restricted to Borda ranking (it is not even clear what Borda ranking means in the context of partial lists, so flexibility here is important) and (2) producing a ranking of all the alternatives in which the Condorcet winner, if one exists, is ranked first and which satisfies the extended criterion as well. We also avoid the NP-hardness of pure Kemeny optimality.

In the context of meta-search we can think of the $\tau_1, \ldots, \tau_k$ as the ordering of the (top, say, 100) pages returned by each of k rank functions in response to a given query (for simplicity, assume the same crawl). Later we will give a formal definition of spam; working now from intuition we have that if a page spams all or even most rank functions, then no combination of these rankings will defeat the spam: garbage in, garbage out. Suppose, however, that a page spams strictly fewer than half the rank functions. In this case the majority of the rank functions (voters) will prefer a "good" page (candidate) to the spam page. In other words, the spam pages are the Condorcet losers, and will occupy the bottom partition of any aggregated ranking that satisfies the extended Condorcet criterion. Similarly, assuming that good pages are preferred by the majority to mediocre ones, these will be the Condorcet winners, and will therefore be ranked highly.

In this setting exact fairness is overkill: there is simply no need for the aggregation procedure to optimally resolve rankings within a partition. If the partition is too coarse, then additional or better rank functions may be called for, but this is not the purview of the aggregation process. Since we expect the top results in μ to be more significant and interesting than the bottom results, our definition of local Kemenization is chosen to facilitate consideration of any prefix of μ.

responses).

Summary of Results and Outline of the Paper Our main results are in Section 2, where we define local Kemeny optimality, consistency, and local Kemenization, prove that any locally Kemeny optimal aggregation of a set $\tau_1, \ldots, \tau_k$ of partial lists satisfies the extended Condorcet criterion, and show how, given any initial aggregation μ of these lists, to obtain a consistent local Kemenization of μ with respect to $\tau_1, \ldots, \tau_k$ in $O(kn \log n)$ time. Section 3 relates our work specifically to meta-search and resistance to search engine spam.

In Section 4 we consider methods for obtaining the initial aggregation μ. We introduce a general technique based on Markov chains together with four concrete suggestions, and also evaluate several established techniques.

Section A (now in the Appendix) obtains new NP-hardness results for achieving Kemeny optimality. In particular, previous work showed hardness in the many-voter-few-candidate setting. We show hardness in the setting of interest in meta-search: many(!) candidates and very few voters. Section B (also in the Appendix) briefly mentions related work in learning theory, sports analysis, and database middleware.

Some experimental results on our proposals are discussed in a companion paper [16].

2 Local Kemeny Optimality and Local Kemenization

We introduce the new notion of a locally Kemeny optimal aggregation, a relaxation of Kemeny optimality that ensures satisfaction of the extended Condorcet criterion and yet is computationally tractable.

Definition 1 (Locally Kemeny optimal) *A permutation π is a* locally Kemeny optimal *aggregation of partial lists $\tau_1, \tau_2, \ldots, \tau_k$ if there is no permutation π' that can be obtained from π by performing a single transposition of an adjacent pair of elements and for which $SK(\pi', \tau_1, \tau_2, \ldots, \tau_k) < SK(\pi, \tau_1, \tau_2, \ldots, \tau_k)$. In other words, it is impossible to reduce the total distance to the τ's by flipping an adjacent pair.*

Note that the above definition is *not* equivalent to requiring that no flipping of any (not necessarily adjacent) pair will decrease the sum of the distances to the τ's.

Example 1: $\pi = (1,2,3), \tau_1 = (1,2), \tau_2 = (2,3), \tau_3 = \tau_4 = \tau_5 = (3,1)$. We have that π satisfies Definition 1, $SK(\pi, \tau_1, \tau_2, \ldots, \tau_5) = 3$, but transposing 1 and 3 decreases the sum to 2.

Every Kemeny optimal permutation is also locally Kemeny optimal, but the converse does not hold (viz., Example 1). Furthermore, a locally Kemeny optimal permutation is not necessarily a good approximation for the optimal. For example, if the τ's are as in Example 1, the number of $(3,1)$ partial lists is very large, and there is only one occurrence of each of the partial lists $(1,2)$ and $(2,3)$, then $(1,2,3)$ is still locally Kemeny optimal, but the ratio (of the SK) to the optimal may be arbitrarily large. Nevertheless, the important observations, proved next, are that a locally Kemeny optimal aggregation satisfies the extended Condorcet property and can be computed efficiently.

Convention. We adopt the convention that π ranks x above y (i.e., prefers x to y) whenever $\pi(x) < \pi(y)$.

Lemma 2 *Let π, a permuation on alternatives $\{1, \ldots, n\}$, be a locally optimal aggregation for partial lists $\tau_1, \tau_2, \ldots, \tau_k$. Then π satisfies the extended Condorcet criterion with respect to $\tau_1, \tau_2, \ldots, \tau_k$.*

Proof: If the lemma is false then there exists partial lists $\tau_1, \tau_2, \ldots, \tau_k$, a locally Kemeny optimal aggregation π, and a partition (T, U) of the alternatives where for all $a \in T$ and $b \in U$ the majority among $\tau_1, \tau_2, \ldots, \tau_k$ prefers a to b, but there are $c \in T$ and $d \in U$ such that $\pi(d) < \pi(c)$. Let (d, c) be a closest (in π) such pair. Consider the immediate successor of d in π, call it e. If $e = c$ then c is adjacent to d in π and transposing this adjacent pair of alternatives produces a π' such that $SK(\pi', \tau_1, \ldots, \tau_k) < SK(\pi, \tau_1, \ldots, \tau_k)$, contradicting the assumption that π is a locally Kemeny optimal aggregation of the τ's. If $e \neq c$ then either $e \in T$, in which case the pair (d, e) is a closer pair in π than (d, c) and also violates the extended Condorcet condition, or $e \in U$, in which case (e, c) is a closer pair than (d, c) that violates the extended Condorcet condition. Both cases contradict the choice of (d, c). $\Box$

The set $\tau_1, \ldots, \tau_k$ of partial lists defines a directed *majority graph G* on the n alternatives, with an edge (x, y) from x to y if a majority of the τ's that contain both x and y rank x above y.

Lemma 3 *Locally Kemeny optimal aggregations of k lists can be computed in $O(kn \log n)$ time.*

Proof: It is not surprising that locally Kemeny optimal aggregations can be found in polynomial time because they are only local minima. A straightforward approach requires $O(n^2)$ time; we describe a technique requiring only $O(kn \log n)$ time (generally, we are interested in the case in which $k \ll n$).

Create the majority graph for $\tau_1, \tau_2, \ldots, \tau_k$ and add anti-parallel edges in the case of a tie. The problem of finding a locally Kemeny optimal aggregation of $\tau_1, \ldots, \tau_k$ is now equivalent to finding a Hamiltonian path in this graph. In [3] it is shown that any comparison–based sorting algorithm can be translated into a Hamiltonian-path finding algorithm in a tournament with the same complexity. (The straightforward algorithm corresponds to insertion sort with no data structures.) Therefore it is possible to find such a path in T in $O(n \log n)$ probes to the edges of T using, for instance, merge sort (the advantage of using merge sort is that the issue of inconsistent answers never arises, which simplifies the execution of the algorithm). The cost of each probe is k accesses to the partial lists (to find out whether there is a majority), so the resulting complexity is $O(kn \log n)$. $\Box$

In Section 3 we explain the value of the extended Condorcet criterion in increasing resistance to search engine spam and in ensuring that elements in the top partitions remain highly ranked. However, specific aggregation techniques may add considerable value beyond simple satisfaction of this criterion; in particular, they may produce good rankings of alternatives within a given partition (as noted above, the extended Condorcet criterion gives no guidance within partition). We now show how, using any initial, not necessarily Condorcet, aggregation μ of partial lists $\tau_1, \ldots, \tau_k$, we can efficiently construct a locally Kemeny optimal aggregation of the τ's that is in a well-defined sense consistent with μ. For example, if the τ's are full lists then μ could be the Borda ordering on the alternatives. Even if a Condorcet winner exists, the Borda ordering may not rank it first. However, by applying our *local Kemenization* procedure (described below), we can obtain a ranking that is maximally consistent with the Borda ordering but in which the Condorcet winners are at the top of the list.

We now formalize our notion of consistency.

Definition 4 *Given partial lists $\tau_1, \tau_2, \ldots, \tau_k$ and a total order μ we say that π is* consistent *with μ and $\tau_1, \tau_2, \ldots, \tau_k$ if $\pi(i) < \pi(j)$ implies that either (a) $\mu(i) < \mu(j)$ or (b) a majority of $\tau_1, \tau_2, \ldots, \tau_k$ prefer i to j (more prefer i over j than j over i, but not necessarily an absolute majority).*

In other words, the order of two elements differs between μ and π *only if* a majority of the τ's support the change (however, consistency does not mandate a switch).

Note that if π is consistent with μ and $\tau_1, \tau_2, \ldots, \tau_k$, then

$$SK(\pi, \tau_1, \tau_2, \ldots, \tau_k) \leq SK(\mu, \tau_1, \tau_2, \ldots, \tau_k),$$

since the only allowed changes decrease the distance to the τ's.

The proof of the next lemma is straightforward from Definition 4.

Lemma 5 *If π is consistent with μ and $\tau_1, \ldots, \tau_k$, then for any $1 \leq \ell \leq n$, if S is the set of ℓ alternatives ranked most highly by μ, the projection of π onto S is consistent with the projections of μ and $\tau_1, \ldots, \tau_k$ onto S.*

As we will see, for any partial lists $\tau_1, \tau_2, \ldots, \tau_k$ and order μ there is a permutation π that is (i) locally Kemeny optimal and (ii) consistent with μ. (Such a π is not necessarily unique.) We will focus particularly on μ-consistent locally Kemeny optimal aggregations that, when restricted to subsets S of the most highly ranked elements in μ, retain their local Kemeny optimality (Definition 6 below). This is desirable whenever we are more sure of the significance of the top results in μ than the bottom ones. In this case the solution is unique (Theorem 7).

Definition 6 *Given partial lists $\tau_1, \tau_2, \ldots, \tau_k$ and a total order μ on alternatives $\{1, 2, \ldots, n\}$, we say that π is a* local Kemenization of μ with respect to $\tau_1, \tau_2, \ldots, \tau_k$ *if (i) π is consistent with μ and (ii) if we restrict attention to the set S consisting of the $1 \leq \ell \leq n$ most highly ranked alternatives in μ, then the projection of π onto S is a locally Kemeny optimal aggregation of the projections of $\tau_1, \ldots, \tau_k$ onto S.*

Theorem 7 *For any partial lists $\tau_1, \tau_2, \ldots, \tau_k$ and order μ on alternatives $\{1, \ldots, n\}$, there exists a* unique *local Kemenization of μ with respect to $\tau_1, \tau_2, \ldots, \tau_k$.*

Proof: We prove the theorem by induction on n, the number of alternatives. The base case $n = 1$ is trivial. Assume the statement inductively for $n - 1$. We will prove it for n. Let x be the last (lowest-ranked) element in μ and let $S = \{1, \ldots, n\} \setminus \{x\}$. Since S is of size $n - 1$ we have by induction that there is a unique permutation σ on the elements in S satisfying the conditions of the theorem. Now insert the removed element x into the lowest-ranked "permissible" position in σ: just below the lowest-ranked element y such that such that (a) no majority among the (original) τ's prefers x to y and (b) for all successors z of y (i.e. $\sigma(y) < \sigma(z)$) there is a majority that prefers x to z. Clearly no two elements of μ were switched unnecessarily and the solution, π, is locally Kemeny optimal from the local Kemeny optimality of σ and the majority properties. Note that the consistency condition requires that x be as low in π as local Kemeny optimality permits, so given σ there is only one place in which to insert x.

Suppose now that μ and $\tau_1, \ldots, \tau_k$ contradict uniqueness: there are two different local Kemenizations of μ with respect to $\tau_1, \ldots, \tau_k$; call them π and π'. If we drop the last element x in μ and let S be as above, then (by property (ii) of local Kemenization) the resulting permutations π_{n-1} and π'_{n-1} must each be local Kemenizations of the restrictions of the τ's to S and (by property (i) and Lemma 5) they must be consistent with the restriction of μ to S. By the induction hypothesis $\pi_{n-1} = \pi'_{n-1}$. As argued above, there is only one place to insert x into this list. $\Box$

The algorithm suggested by this proof may take $O(n^2 k)$ time in the worst case (say a transitive tournament where μ is the anti-transitive order). However, in general it requires time proportional to the K-distance between μ and the solution. We do not expect μ to be uncorrelated with the solution and therefore anticipate better performance in practice.

Summary: Our Approach to Rank Aggregation. We now have all the components of our new approach to rank aggregation:

> Given $\tau_1, \ldots, \tau_k$, use your favorite aggregation method to obtain a permutation μ. Output the (unique) local Kemenization of μ with respect to $\tau_1, \ldots, \tau_k$.

Such an approach preserves the strengths of the initial aggregation μ, while efficiently ensuring satisfaction of the extended Condorcet criterion. In particular, the Condorcet losers receive low rank, while the Condorcet winners receive high rank. Although motivated by questions in meta-search, this new approach to rank aggregation is of independent interest.

3 Spam and Condorcet Losers

Intuitively, a ranking function has been spammed by a page in a database of pages if for a given query it ranks the page "too highly" with respect to other pages in the database, in the view of a "typical" user. Indeed, in accord with this intuition, search engines are both rated [30, 31] and trained by human *evaluators*. This approach to defining spam: (1) permits an author to raise the rank of her page by improving the content; (2) puts *ground truth* about the relative value of pages into the purview of the users—in other words, the definition does not assume the existence of an absolute ordering that yields the "true" relative value of a pair of pages on a query; (3) does not assume unanimity of users' opinions; and (4) suggests some natural ways to automate training of engines to incorporate useful biases, such as geographic bias.

Formally, we assume a pool $\mathcal{E} = \{E_1, \ldots, E_m\}$ of *evaluators*, where each E_i takes as input a pair of pages (p, p') and a query Q and defines a partial order by replying with a value in $\{+1, -1, 0\}$. A *ranking function* f takes as input a database D of web pages and a query Q, and produces a rank ordering on the pages in D. Then f is *unspammable* with respect to $\mathcal{E}$ if $\forall D \, \forall p, p' \in D \, \forall Q$ at least as many evaluators agree with the relative ranking of p and p' produced by $f(Q, D)$ as disagree.

We believe that reliance on evaluators in defining spam is unavoidable. If the evaluators are human, the typical scenario during the design and training of search engines, then the eventual product will incorporate the biases of the training evaluators.

Amplification of Spam Resistance. Fix a set $\mathcal{E}$ of evaluators. In the sequel, whenever we say "unspammable" we mean "unspammable with respect to $\mathcal{E}$." We say that a ranking function f is ε-*weakly-unspammable* on a database D of pages if for all queries Q, for a randomly selected pair of pages (p', p), the probability that f's relative ranking of this pair of pages on query Q agrees with the majority of the evaluators' opinions is at least $1/2 + \varepsilon$. We say that f is ε-*unspammable* if this probability is at least $1 - \varepsilon$.

Assume ε-weakly unspammable ranking functions return partial lists $\tau_1, \ldots, \tau_k$ in response to a query Q. In the sequel, let G_R be the majority graph formed from the outputs of the *ranking functions* (the τ's), and let G_E denote the majority graph constructed according to the majority opinions of the *evaluators*. Ideally, we would like to be able to make statements of the following form:

1. If for all (but an ε fraction of) pairs of pages p, p' a majority of the rank functions agree with a majority of the evaluators on the relative ranks of p and p', then the aggregation will be ε-unspammable; or

2. Under the same assumption as above and assuming a fixed query, letting AF denote some particular aggregation function, if the set X of alternatives can be partitioned into $\bar{S}, S$ (intuitively, not spam and spam, respectively) such that $\forall \bar{s} \in \bar{S}, s \in S : G_E(\bar{s}, s) = 1$, then for some "nice" function f, $\Pr_{\bar{s} \in \bar{S}, s \in S}[AF(\bar{s}) < AF(s)) = 1] \geq 1 - f(\varepsilon)$. In other words, the aggregation function should (almost) observe the not spam/spam partition defined by the evaluators.

Before showing that some version of Statement 2 actually holds, we note two limiting factors. First, the G_E graph may have cycles. For example, the graph may be a directed triangle. In this case any rank ordering on the (three) alternatives will disagree with a majority of the evaluators with probability at least $1/3$ on a randomly chosen pair of vertices. This example rules out Statement 1.

Second, consider an extended Condorcet partitioning of the vertices of the G_E graph. The ε fraction of the pairs (p, p') for which G_R and G_E *disagree* may contain the $\bar{S}, S$ cut in the G_E graph. In this case, even though G_R and G_E agree on a randomly chosen edge with probability $1 - \varepsilon$, the roles of spam and non-spam could be reversed, with spam consistently outranking all non-spam.

The solution is to restrict attention to the case in which whenever $G_E(x, y) = 1$ (a majority of the evaluators expressing a preference prefers x to y) $G_R(x, y) = 1$ (in a majority of the lists containing both x and y x is ranked above y). This is reasonable: if for some pair of pages a majority of the engines are spammed, the aggregation function is working with overly bad data (garbage in, garbage out).

In this case any extended Condorcet partitioning consistent with the G_R graph will also be consistent with the G_E graph. In particular, for any partitioning of X into $X_1, \ldots, X_p$, such that for all $a < b$, for all $\bar{s} \in X_a$, $s \in X_b$, $G_E(\bar{s}, s) = 1$ it is also the case that $G_R(\bar{s}, s) = 1$, and so by Lemma 2 any locally Kemeny optimal aggregation of the τ's will rank $\bar{s}$ above s. Thus, the spam pages are the Condorcet losers, and will be pushed to the bottom during local Kemenization.

4 Some Initial Aggregation Methods

Recall that our approach to rank aggregation is to first apply any initial favorite rank aggregation method to obtain a ranking μ, and then to locally Kemenize it. We propose the use of Markov chains for constructing the initial aggregations μ. We suggest and evaluate some specific Markov chains for this purpose, and relate them to previously known methods. We focus on the following criteria: relationship to Kemeny optimal solutions, computational complexity, and suitability for certain cases of incomplete information (either the ranking functions have incomplete knowledge of the alternatives or the τ's are truncated). We generalize Young's result [40] that "positional" methods cannot satisfy the (original) Condorcet criterion.

4.1 Proposal: Markov Chains

We propose a general method for obtaining an initial aggregation of partial lists, using Markov chains. The states of the chain correspond to the n candidates to be ranked, the transition probabilities depend in some particular way on the given (partial) lists, and the *stationary probability distribution* will be used to sort the n candidates to produce the aggregation. There are several motivations for this approach:

(1) Handling partial lists and "top d" orderings: In this case each τ_i contains only the "top d" elements in i's preference list. There is no way to know if a given alternative (page) not appearing in τ_i is known to i (and assigned a low rank) or is simply unkown to i. Rather than require every pair of pages (candidates) i and j to be compared by every search engine (voter), we may now use the the *available* comparisons between i and j to determine the transition probability between i and j, and exploit the connectivity of the

chain to (transitively) "infer" comparison outcomes between pairs that were not explicitly ranked by any of the search engines. The intuition is that Markov chains provide a more holistic viewpoint of comparing all n candidates against each other—significantly more meaningful than ad hoc and local inferences like "if a majority prefer A to B and a majority prefer B to C, then A should be better than C."

(2) Handling uneven comparisons: If a web page P appears in the bottom half of about 70% of the lists, and is ranked Number 1 by the other 30%, how important is the quality of the pages that appear on the latter 30% of the lists? If these pages all appear near the bottom on the first set of 70% of the lists and the winners in these lists were not known to the other 30% of the search engines that ranked P Number 1, then perhaps we shouldn't consider P too seriously. In other words, if we view each list as a tournament within a league, we should take into account the strength of the schedule of matches played by each player. This issue has attracted much attention with respect to the ranking of sports teams [26, 38] (see Appendix B.1). The Markov chain solutions we discuss are similar in spirit to the approaches considered in the mathematical community for this problem (eigenvectors of linear maps, fixed points of nonlinear maps, etc.).

(3) Enhancements of other heuristics: Heuristics for combining rankings are motivated by some underlying principle. For example, Borda's method is based on the idea "more wins is better." This gives some figure of merit for each candidate. It is natural to extend this and say "more wins against *good* players is even better," and so on, and iteratively refine the ordering produced by a heuristic. In the context of web searching, the HITS algorithm of Kleinberg [27] and the PageRank algorithm of Brin and Page [8] are motivated by similar considerations. As we will see, some of the chains we propose are natural extensions (in a precise sense) of Borda's method, sorting by geometric mean, and Copeland's method (see Section 4.2).

(4) Computational efficiency: In general, setting up one of these Markov chains and determining its stationary probability distribution takes about $\Theta(n^2k + n^3)$ time. However, in practice, if we explicitly compute the transition matrix in $O(n^2k)$ time, a few iterations of the power method will allow us to compute the stationary distribution. In fact, we suggest an even faster method for practical purposes. For all of the chains that we propose, with about $O(nk)$ (linear in input size) time for preprocessing, it is usually possible to simulate one step of the chain in $O(k)$ time; thus by simulating the Markov chain for about $O(n)$ steps, we should be able to sample from the stationary distribution pretty effectively. This is usually sufficient to identify the top few candidates in the stationary distribution in $O(nk)$ time, perhaps considerably faster in practice.

We now propose some specific Markov chains, denoted MC_1, MC_2, MC_3 and MC_4.

MC_1: If the current state is page P, then the next state is chosen uniformly from the multiset of all pages that were ranked higher than (or equal to) P by some search engine that ranked P, that is, from the multiset $\bigcup_i \{Q \mid \tau_i(Q) \leq \tau_i(P)\}$.

MC_2: If the current state is page P, then the next state is chosen as follows: first pick a ranking τ uniformly from all the partial lists $\tau_1, \ldots, \tau_k$ containing P, then pick a page Q uniformly from the set $\{Q \mid \tau(Q) \leq \tau(P)\}$.

MC_3: If the current state is page P, then the next state is chosen as follows: first pick a ranking τ uniformly from all the partial lists $\tau_1, \ldots, \tau_k$ containing P, then uniformly pick a page Q that was ranked by τ. If $\tau(Q) < \tau(P)$ then go to Q, else stay in P.

MC_4: If the current state is page P, then the next state is chosen as follows: first pick a page Q uniformly from the union of all pages ranked by the search engines. If $\tau(Q) < \tau(P)$ for a *majority* of the lists τ that ranked both P and Q, then go to Q, else stay in P.

Of these four Markov chains, MC_1 is the most straightforward: the idea is that in each step, we move from the current page to a better page, allowing about $1/j$ probability of staying in the same page, where j is roughly the average rank of the current page. MC_2 is the only one among these that takes into account the fact that we have several *lists* of rankings, not just a collection of pairwise comparisons among the pages. As a consequence, MC_2 is arguably the most representative of minority viewpoints of sufficient statistical significance; it also protects specialist views.

Recall that if X is a set of n candidates, and $\tau_1, \tau_2, \ldots, \tau_k$ are full (ranked) lists of the candidates, then for each candidate c and list τ_i, Borda's method first assigns a score $B_i(c) =$ the number of candidates ranked below c in τ_i, and then the total Borda score $B(c)$ is defined as $\sum_i B_i(c)$. The candidates are then sorted in decreasing order of total Borda score. Thus Borda's method is a "positional" method, in that it assigns a score corresponding to the positions that a candidate appears in within each voter's ranked list of preferences, and the candidates are sorted by their total score. Lemma 8 says that MC_2 generalizes the geometric mean analogue of Borda's method.

Lemma 8 *For full lists, if the initial state is chosen uniformly at random, after one step of MC_2, the distribution induced on its states produces a ranking of the pages such that P is ranked higher than Q iff the* geometric mean *of the ranks of P is lower than the geometric mean of the ranks of Q.*

Lemma 9 *For full lists, if the initial state is chosen uniformly at random, after one step of MC_3, the distribution induced on its states produces a ranking of the pages such that P is ranked higher than Q iff the Borda score of P is higher than the Borda score of Q.*

Thus, MC_3 is a generalization of Borda's method. This is natural, considering that in any state P, the probability of staying in P is roughly the fraction of pairwise contests (with all other pages) that P won—a very Borda-like measure. Similarly, MC_4 generalizes Copeland's method of sorting all candidates by the number of pairwise majority contests they won (see Section 4.2).

We have constructed several examples that differentiate the behavior of MC_1, MC_2, and MC_3, and examples showing that the stationary distributions of these chains do not satisfy the extended Condorcet principle.

A technical point to note while using Markov chains for ranking: The strongly connected components of the underlying graph of the Markov chain may be described by a directed acyclic graph. If this DAG has a sink node, then the stationary distribution of the chain will be entirely concentrated in the strongly connected component corresponding to the sink node. In this case, we only obtain an ordering of the alternatives present in this component; if this happens, the natural way to proceed is to remove these states from the chain and repeat the process to rank the remaining nodes. Of course, if this component has sufficiently many alternatives, one may stop the aggregation process and output a partial list containing some of the best alternatives. If the DAG of connected components is (weakly) connected and has more than one sink node, then we will obtain two or more clusters of alternatives, which we could sort by the total probability mass of the components. If the DAG has several weakly connected components, we will obtain incomparable clusters of alternatives.

4.2 Positional Methods

Given k full lists $\tau_1, \tau_2, \ldots, \tau_k$, Borda's method can be thought of as assigning a k-element *position* vector to each candidate (the positions of the candidate in the k lists), and sorting the candidates by the L_1 norm of these vectors. Of course, there are plenty of other possibilities with such position vectors: sorting by L_p norms for $p > 1$, sorting by the median of the k values, sorting by the geometric mean of the k values, etc. A primary advantage of positional methods is that they are computationally very easy: they can be implemented in linear time on a RAM (i.e., in $O(nk)$ time for k full lists of n candidates). They also enjoy several nice and natural properties (called anonymity, neutrality and consistency in the social choice literature) [40]. However, Theorem 10 (proof in Appendix C) shows that they cannot satisfy the Condorcet criterion (see also [40]). Another major limitation is their (in)extensibility to partial lists and top-d lists. It has been proposed (e.g., in an article that appeared in *The Economist* [36]) that the right way to extend Borda to partial lists is by apportioning all the excess score equally among all unranked candidates. This idea stems from the goal of being *unbiased*; however, it is easy to show that for any method of assigning scores to unranked candidates, there are parital information cases in which undersirable outcomes occur. This is unfortunate since positional approaches have some statistical benefits: for example, sorting by geometric mean has the advantage of being fairly representative of minority viewpoints, and sorting by median is closely related to optimizing total distance under the Spearman's *footrule* metric on the space of permutations, discussed below.

Theorem 10 *A set of permutations $\tau_1, \ldots, \tau_k$ on alternatives $\{1, \ldots, n\}$ defines a k-element position vector v_i for each alternative i as follows: $v_i(j) = c$ if i is the cth element in τ_j. Let $f : \mathbb{N}^k \to \mathbb{R}$ be any one-to-one function. Then the aggregation π obtained by ordering alternatives according to $f(v_i)$ does not satisfy the Condorcet criterion: there exist $\tau_1, \ldots, \tau_k$ for which the f-based aggregation π will not rank the Condorcet winner first.*

Other Distance Measures. It is natural to ask if distance measures other than the K-distance are relevant to our problem of aggregating rankings from search engines, and to what extent they satisfy the various properties that are desirable in our context (see [14] for a discussion of several metrics). The one we find most relevant is *Spearman's footrule distance*, defined between permutations σ and π by $F(\sigma, \pi) = \sum_j |\sigma(j) - \pi(j)|$. This distance is fairly natural for our purpose: it measures the displacement of each page between the two rankings σ and π. It turns out that the footrule distance between any two permutations approximates their Kendall-tau distance to within factor two [15]: $K(\sigma, \pi) \leq F(\sigma, \pi) \leq 2K(\sigma, \pi)$. Consequently, any algorithm that computes a footrule optimal aggregation is automatically a 2-approximation algorithm for finding Kemeny optimal permutations[2]. We can construct examples where this approximation is tight. On the positive side, we prove:

Lemma 11 *If the median positions of the candidates in the lists $\tau_1, \ldots, \tau_k$ form a permutation, then this permutation is a footrule optimal aggregation.*

Spearman's rank correlation measure [14] is defined by $S(\sigma, \pi) = \sum_j (\sigma(j) - \pi(j))^2$. Thus, $K(\sigma, \pi) \leq \sqrt{n} S(\sigma, \pi) \leq 2\sqrt{n} K(\sigma, \pi)$. In the full paper we show:

Lemma 12 *If the Borda scores of candidates are unique, then the permutation implied by the Borda scores minimizes the sum of the Spearman rank correlations $\sum_i S(\sigma, \tau_i)$.*

More generally (e.g., when the median positions and Borda scores do not necessarily determine a permutation) in Appendix C we prove:

Theorem 13 *Footrule-optimal and Spearman's rank correlation-optimal aggregations of full lists can be computed in polynomial time, specifically, the time needed to find a minimum cost perfect matching in a bipartite graph.*

The computation of a footrule-optimal aggregation for partial lists is more problematic. Given two partial lists (or a partial list and a full list) σ and π, one could define the F-distance $F(\sigma, \pi)$ between them as the Spearman footrule distance between the projections of σ and π to their intersection. Given k partial lists $\tau_1, \tau_2, \ldots, \tau_k$, one could then ask for a permutation π on X, the union of the elements in the τ_i's, such that $\sum_i F(\pi, \tau_i)$ is minimized. It is easy to see that if the τ_i's are lists of two elements each, then $F(\pi, \tau_i)$ and $K(\pi, \tau_i)$ are either both zero, or $F(\pi, \tau_i) = 2$ and $K(\pi, \tau_i) = 1$; thus $\sum_i F(\pi, \tau_i) = 2 \sum_i K(\pi, \tau_i)$, and therefore, finding a footrule optimal aggregation and Kemeny optimal aggregation are equivalent. We observe in Appendix A that this special case is equivalent to the NP-hard problem of computing minimum feedback edge sets.

[2] Such approximations in themselves are not difficult to find, since in the case of k full lists, K is a metric and hence one of the τ's is a $2(1 - 1/k)$-approximation to the L_1 minimum (we thank Jon Kleinberg for pointing this out).

Copeland's method [12] is an attractive method for full lists: sort the candidates by the number of pairwise majority wins minus pairwise majority losses. This amounts to sorting the nodes in the majority graph by outdegree minus indegree. Copeland's method satisfies the extended Condorcet condition, can be computed in $O(n^2k)$ time and is generalized by MC_4.

5 Conclusions and open questions

We have proposed a new approach to the very old problem of rank aggregation: compute any favorite initial aggregation μ and then compute the local Kemenization of μ. By identifying spam with Condorcet losers we can reduce search engine spam; the approach is particularly useful when the different ranking functions are operating on the same database of pages.

This work was inspired by conversations with Helen Nissenbaum concerning her research on bias in computing systems [21, 25]. As noted above, if a search engine is trained using human evaluators, then the engine will reflect the biases of the trainers. We have not given a formal definition of bias, but this observation clearly indicates a need for aggregation functions that protect minority/specialist views. The term "bias" is emotionally loaded. Indeed, certain kinds of biases, such as geographic bias, are clearly desirable: when we ask the search engine to find movie theaters, we typically want movie theaters that are within driving distance. This work suggests personalization as an avenue to pursue. It also motivates the construction of specialized search engines. This is a rich area for future research.

Questions not specific to meta-search include: When can sampling be used to improve performance? How can we achieve even-handed aggregation; that is, if there exists a set of good solutions, say, locally Kemeny optimal aggregations, how can we ensure that the chosen output is "fair," for example, minimizes them like maximum disagreement with any one of the input sequences?

Several questions specific to meta-search surround the indistinguishability to the aggregation procedure of the following two cases: (1) the search engine does not have a certain page P in its database and (2) the engine has P in its database but ranks P below the top d pages (in response to a given query). In the first case it is possible that P would have received a high rank, and in the second P receives a low rank; these should lead to different aggregation results, but the information is simply not present. Are there (realistic) modifications to the web page interface that would resolve this ambiguity? Can/should crawling be separated from searching? Already there are web sites that return different pages to different search engines following "reverse engineering" of the different rank functions. How can this be addressed in meta-search?

Finally, our several proposals require experimental analysis. Preliminary results are discussed in a companion paper [16].

References

[1] M. Adler, P. Gemmell, M. Harchol-Balter, R. Karp, and C. Kenyon. Selection in the presence of noise: The design of playoff systems. *Proc. 5th Symposium on Discrete Algorithms*, pp. 564–573, 1994.

[2] A. Andersson and O. Peterssoni. Approximate indexed lists. *J. Algorithms*, 29:256–276, 1998.

[3] A. Bar-Noy and J. Naor. Sorting, minimal feedback sets, and Hamiltonian paths in tournaments. *SIAM J. on Discrete Mathematics*, 3 (1):7–20, 1990.

[4] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[5] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[6] D. Black. *The Theory of Committees and Elections.* Cambridge University Press, Cambridge, 1958.

[7] J. C. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[9] Center for Voting and Democracy Factsheet. http://www.igc.apc.org/cvd/frames/java2000.htm

[10] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *J. of Artificial Intelligence Research*, 10:243–270, 1999.

[11] M.-J. Condorcet. *Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*, 1785.

[12] A. H. Copeland. A reasonable social welfare function. *Mimeo*, University of Michigan, 1951.

[13] L. F. Cranor. *Declared-Strategy Voting: An Instrument for Group Decision-Making.* Ph. D. Thesis, Washington University, 1996. `http://www.research.att.com/~lorrie/pubs/diss/`

[14] P. Diaconis. *Group Representation in Probability and Statistics*. IMS Lecture Series 11, Institute of Mathematical Statistics, 1988.

[15] P. Diaconis and R. Graham. Spearman's footrule as a measure of disarray. *J. of the Royal Statistical Society, Series B*, 39(2):262–268, 1977.

[16] C. Dwork, R. Kumar, M. Naor, D. Sivakumar. Rank Aggregation Methods for the Web. *Manuscript.*

[17] G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[18] R. Fagin. Combining Fuzzy information from multiple systems. *JCSS*, 58(1):83–99, 1999.

[19] R. Fagin, M. Franklin, A. Lotem, and M. Naor, Optimal Aggregation Algorithms for Middleware. *manuscript*, 2000

[20] P. C. Fishburn. Condorcet social choice functions. *SIAM J. on Applied Mathematics*, 33(3):269–489, 1977.

[21] B. Friedman and H. Nissenbaum. Bias in computer systems. *ACM Transactions on Information Systems*, 14(3):330–347, 1996.

[22] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Proc. 15th International Conference on Machine Learning*, 1998.

[23] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Applied Statistics*, 48:377–394, 1999.

[24] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson Elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Proc. International Colloquium on Automata, Languages, and Programming*, pp:214–224, 1997.

[25] L. Introna and H. Nissenbaum. Shaping the Web: why the politics of search engines matters. *to appear, The Information Society 16*, pp. 1–17, 2000.

[26] J. P. Keener. The Perron-Frobenius theorem and the rating of football teams. *SIAM Review*, 35(1):80–93, 1993.

[27] J. Kleinberg. Authoritative sources in a hyperlinked environment. *J. of the ACM*, 46(5):604–632, 1999.

[28] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.

[29] A. Lijphart. *Electoral Systems and Party Systems: A Study of Twenty-Seven Democracies 1945,1990.* Oxford University Press, Oxford, 1994.

[30] Media Metrix search engine ratings.
`http://www.searchenginewatch.com/reports/mediametrix.html`

[31] Nielsen/NetRatings search engine ratings.
`http://www.searchenginewatch.com/reports/netratings.html`

[32] D. M. Pennock and E. Horvitz. Analysis of the axiomatic foundations of collaborative filtering. *Workshop on AI for Electronic Commerce at the 16th National Conference on Artificial Intelligence*, 1999.

[33] D. M. Pennock, E. Horvitz, and C. Lee Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. *Proc. of the 17th National Conference on Artificial Intelligence*, 2000.

[34] D. M. Pennock, P. Maynard-Reid II, C. Lee Giles, and E. Horvitz. A normative examination of ensemble learning algorithms. *Proc. 17th International Conference on Machine Learning*, 2000.

[35] D. G. Saari. *Basic Geometry of Voting*. Springer-Verlag, 1995.

[36] D. G. Saari. The mathematics of voting: Democratic symmetry. *Economist*, pp. 83, March 4, 2000.

[37] W. D. Smith. Learning and rating systems.
http://www.neci.nj.nec.com/homepages/wds/ratingspap.ps

[38] M. Stob. A supplement to 'A mathematician's guide to popular sports'. *American Mathematical Monthly*, 91(5):277–282, 1984.

[39] M. Truchon. An extension of the Condorcet criterion and Kemeny orders. *cahier 98-15 du Centre de Recherche en Économie et Finance Appliqueés*, 1998.

[40] H. P. Young. An axiomatization of Borda's rule. *J. Economic Theory*, 9:43–52, 1974.

[41] H. P. Young. Condorcet's Theory of Voting. *American Political Science Review*, 82: 1231–1244, 1988.

[42] H. P. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM J. on Applied Mathematics*, 35(2):285–300, 1978.

[43] E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29:436–460, 1926.

A Complexity of Kemeny optimal

In this section, we study the complexity of finding a Kemeny optimal permutation. We show that computing a Kemeny optimal permutation is NP-hard, even when the input consists of four full lists $\tau_1, \tau_2, \tau_3, \tau_4$. For partial lists of length 2 finding Kemeny optimal solution is exactly the same problem as finding a minimum feedback arc set, and hence is NP-hard (see [17] for approximation results). The problem is also known to be NP-hard for an unbounded number of complete lists [4] (see [24] for other complexity results).

We remark that computing a Kemeny optimal permutation for two lists is trivial—simply output one of the input lists. The complexity of computing a Kemeny optimal permutation for three full lists is open; we show later in this section that this problem is reducible to the problem of finding minimum feedback edge sets on tournament graphs, which, as far as we know, is open as well.

Computing a Kemeny optimal permutation for an unbounded number of *partial* lists is easily seen to be NP-hard by a straightforward encoding of the feedback edge set problem: for each edge (i, j), create a partial list of two elements: i followed by j.

Theorem 14 *The problem of computing a Kemeny optimal permutation for a given collection of (full) lists $\tau_1, \tau_2, \ldots, \tau_k$, for even integers $k \geq 4$, is NP-hard. The corresponding decision problem is NP-complete.*

Proof: The reduction is from the feedback edge set problem. Given a directed graph $G = (V, E)$, and an integer $L \geq 0$, the question is whether there exists a set $F \subseteq E$ such that $|F| \leq L$ and $(V, E - F)$ is acyclic. Let $n = |V|$ and $m = |E|$. Given G, we first produce a graph $G' = (V', E')$ by "splitting" each edge of G into two edges; formally, $V' = V \cup \{v_e \mid e \in E\}$ and $E' = \{(i, v_{i,j}), (v_{i,j}, j) \mid (i, j) \in E\}$. The easy fact that we will use later is that G has a feedback edge set of size L if and only if G' does.

Arbitrarily order all the vertices of G' so that the vertices in V receive the numbers $1, \ldots, n$ (and the vertices of the form v_e receive numbers $n+1, \ldots, n+m$). Whenever we wish to refer to this ordering, we will denote it by $\mathcal{O}$. For a vertex $i \in V$, let *Out*(i) denote a listing of the out-neighbors of i in G' in the order prescribed by $\mathcal{O}$; similarly let *In*(i) denote the in-neighbors of i in G' in the order prescribed by $\mathcal{O}$. Note that none of the lists *Out*(i) or *In*(i) contains any vertex from the original graph G. We now define four full lists on the set V'. For a list $\mathcal{L}$, the notation $\mathcal{L}^R$ denotes the reversal of the list.

$$\begin{array}{lllllllll}
\tau_1 & = 1, & \mathit{Out}(1), & 2, & \mathit{Out}(2), & \ldots, & n, & \mathit{Out}(n) \\
\tau_2 & = n, & \mathit{Out}(n)^R, & n{-}1, & \mathit{Out}(n{-}1)^R, & \ldots, & 1, & \mathit{Out}(1)^R \\
\tau_3 & = 1, & \mathit{In}(1), & 2, & \mathit{In}(2), & \ldots, & n, & \mathit{In}(n) \\
\tau_4 & = n, & \mathit{In}(n)^R, & n{-}1, & \mathit{In}(n{-}1)^R, & \ldots, & 1, & \mathit{In}(1)^R.
\end{array}$$

The idea is that in τ_1, each vertex in V precedes all its out-neighbors in G', but the ordering of the out-neighbors of a vertex, as well as the ordering of the vertex-neighbor groups are arbitrary (according to $\mathcal{O}$). The list τ_2 "cancels" the effect of this arbitrariness in ordering the neighbors of a vertex and the vertex-neighbor groups, while "reinforcing" the ordering of each vertex in V above its out-neighbors in G'. Similarly, in τ_3 and τ_4, each vertex of the original vertex set V is preceded by its in-neighbors in G', with suitably arranged cancellations of the artificial ordering among the other pairs.

The main claim is that G has a feedback edge set of size L if and only if there is a permutation π such that $\sum_r K(\pi, \tau_r) \leq L'$, where $L' = 2L + 2(\binom{n}{2} + \binom{m}{2} + m)$.

First suppose that G has a feedback edge set F of size L. It is easy to see that the set $F' = \{(i, v_{i,j}) \mid (i,j) \in F\}$ is a feedback edge set of G', and $|F'| = L$. The graph $(V', E' - F')$ is acyclic, so by topologically sorting the vertices of this graph, we obtain an ordering π of the vertices in V' such that for every $(i,j) \in E' - F'$, i is placed before j in π. We claim that π is an ordering that satisfies $\sum_r K(\pi, \tau_r) \leq 2L + 2(\binom{n}{2} + \binom{m}{2} + m)$. Note that regardless of how π was obtained, the last three terms are inevitable:

(1) for each pair $i, j \in V$, exactly one of τ_1 and τ_2 places i above j and the other places j above i, so there is a contribution of 1 to $K(\pi, \tau_1) + K(\pi, \tau_2)$; similarly, there is a contribution of 1 to $K(\pi, \tau_3) + K(\pi, \tau_4)$. This accounts for the term $2\binom{n}{2}$.

(2) a similar argument holds for pairs $v_e, v_{e'}$, and there are $\binom{m}{2}$ such pairs, accounting for the term $2\binom{m}{2}$.

(3) a similar argument holds for pairs $v_{i,j}, j$ with respect to τ_1 and τ_2, and for pairs $i, v_{i,j}$, with respect to τ_3 and τ_4. The total number of such pairs is $2m$.

The only remaining contribution to the total distance of π from the τ's comes from the $(i, v_{i,j})$ pairs with respect to τ_1 and τ_2 (where i precedes $v_{i,j}$ in both lists), and the $(v_{i,j}, j)$ pairs with respect to τ_3 and τ_4 (where $v_{i,j}$ precedes j in both lists). Of these, a pair contributes 2 to the total Kemeny distance $\sum_r K(\pi, \tau_r)$ precisely if it occurs as a "back edge" with respect to the topological ordering π of the vertices of G'; since $(V', E' - F')$ is acyclic, the total number of such back edges is at most $|F'| = L$.

Conversely, suppose that there exists a permutation π that achieves a total Kemeny distance of at most $L' = 2L + 2(\binom{n}{2} + \binom{m}{2} + m)$. We have already argued (in items (1), (2), and (3) above) that π must incur a distance of $2(\binom{n}{2} + \binom{m}{2} + m)$ with respect to the τ's, the so the only extra distance between π and the τ's comes from pairs of the form $(i, v_{i,j})$ in τ_1 and τ_2, and of the form $(v_{i,j}, j)$ in τ_3 and τ_4. Once again, each such pair contributes either 0 or 2 to the total distance. Consider the pairs that contribute 2 to the distance, and let the corresponding set of edges in E' be denoted by F'. Now, $(V', E' - F')$ is acyclic since every edge that remains in $E' - F'$, by definition, respects the ordering in π. Thus F' is a feedback edge set of G' of size at most L', and the set $F = \{(i,j) \mid (i, v_{i,j}) \in F' \vee (v_{i,j}, j) \in F'\}$ is a feedback edge set of G of size at most L'.

This completes the proof that computing a Kemeny optimal permutation is NP-hard even when the input consists of four full lists. The proof for the case of even k, $k > 4$, is a simple extension: first produce four lists as above, then add $(k-4)/2$ pairs of lists σ, σ^R, where σ is an arbitrary permutation. This addition clearly preserves Kemeny optimal solutions; the distance parameter is increased by an additive $((k-4)/2)\binom{n+m}{2}$ term. $\square$

A curious feature of Theorem 14 is that the number of input lists is always even. Intuitively, this is due to the following reason: Given a graph $G = (V, E)$, suppose a certain pair (u, v) of vertices does not have an edge in either direction. In the instance of Kemeny optimal permutation that we create, we do not wish to introduce any preference for any one of u, v to be ranked above the other. With an even number of lists, we can easily achieve the necessary "cancellations." However, if we produce an odd number of lists, there is a well-defined "majority preference" for ranking one over the other, and this makes the relationship between a Kemeny optimal permutation and a minimum feedback edge set unclear.

Following the above reasoning, it is natural to ask if the case of odd number of lists is related to the minimum feedback edge set problem on tournament graphs (where every pair of vertices have exactly one directed edge between them). We show that this is indeed the case:

Theorem 15 *(1) The problem of computing a Kemeny optimal permutation for k full lists, where k is an odd integer, is reducible to the problem of computing a minimum feedback edge set on weighted tournament graphs with weights between 1 and $k-2$. In particular, computing a Kemeny optimal permutation for 3 full lists reduces to the minimum feedback edge set problem on unweighted tournaments.*

(2) The problem of computing a minimum weight feedback edge set on weighted tournament graphs with polynomially bounded weights is NP-hard;

(3) The problem described in part (2) is reducible to computing a Kemeny optimal permutation for an (non-constant) odd number of full lists; thus the latter is NP-hard.

Remark 3. As far as we know, the problems of Kemeny on bounded odd number of lists and min. weight feedback edge sets on tournaments with bounded edge weights are neither known to be polynomial-time solvable nor known to be NP-hard. We find it strange that the minimum feedback edge set problem on unweighted and bounded weight tournaments—arguably a very natural problem in the context of round robin sports tournaments—has not received much attention from the viewpoint of algorithms/complexity.

Proof: (of Theorem 15) For part (1), given an odd number of lists $\tau_1, \tau_2, \ldots, \tau_k$ on n objects, consider the n-vertex graph G defined as follows: For each pair $u \neq v$, there is a directed edge from u to v iff the majority of the τ_i's rank u higher than v. The weight on this edge is the number of τ_i's that rank u higher minus the number of τ_i's that rank v higher.

The graph G thus defined is a tournament with maximum edge weight k, and it is easy to see that G has a feedback edge set of total weight L iff there is a permutation of total K-distance L from the τ_i's. To achieve the advertised bound $k-2$ on the maximum edge weight, we make the following sequence of observations:

(1) In a tournament, for every set of $r > 3$ vertices $u_1, u_2, \ldots, u_r$, that form a directed cycle u_1–u_2–$\ldots$–u_r–u_1, there is a subset of these vertices that form a directed triangle. (proof: if (u_3, u_1) is an edge, then u_1–u_2–u_3–u_1 is a directed triangle; if (u_1, u_3) is an edge, then we have a shorter cycle u_1–u_3–u_4–$\ldots$–u_1, and we can repeat the process.)

(2) Therefore, in tournament graphs, every minimum weight feedback edge set is a minimum weight subset of edges that intersects every triangle.

(3) In the graph G constructed above (from $\tau_1, \tau_2, \ldots, \tau_k$), an edge of weight k cannot be part of any triangle in G. (proof: suppose (u, v) is an edge of weight k; by definition, every τ_i ranks u above

v. Now suppose the majority of them rank v above w; then it is impossible for a majority to rank w above u.)

(4) Therefore, any minimal set of edges that intersects every triangle in G does not include an edge of weight k.

(5) Consequently, we may change every weight-k edge into a weight-1 edge; the next highest possible weight is $k-2$.

We now turn to part (2) of the Theorem. Start with an instance of the minimum feedback edge set problem—a directed graph $G=(V,E)$ and an integer L. Convert this to a weighted tournament G' as follows. Let $m=|E|$, and let $z>m$ be any integer. Assign a weight of z to every edge in E. If a pair (u,v) has no edge between them (in either direction) in G, then create an arbitrarily directed edge between them of weight 1. In the tournament thus formed, the newly added edges could form cycles that did not exist before, but the set of newly added edges are sufficient to cover all such cycles, and the total weight of all these edges is at most m, which is less than the weight of any one of the original edges. Furthermore, removing the new edges will not break any cycle that was present in G, so any feedback edge set of G', minus the new edges, is a feedback edge set of G. Thus G' has a feedback edge set of weight at most $zL+m$ iff G has a feedback edge set of weight at most L.

For part (3) of the Theorem, for simplicity, we first consider the case of unweighted tournaments. Given a tournament $G=(V,E)$, we let the first list τ_1 be an arbitrary permutation, say the list $1,2,\ldots,n$. Let $U\subseteq E$ be the set of ("unhappy") edges $(u,v)\in E$ such that τ_1 ranks v higher than u. For each edge $e=(u,v)\in U$, we add two lists τ_e,τ'_e, defined as follows: $\tau_e=u,v,Z$, $\tau'_e=Z^R,u,v$, where Z is an arbitrary ordering of $V-\{u,v\}$. If $h=|U|$, the total number of lists created is $2h+1$. It is easy to check that for every pair (u,v), $(u,v)\in E$ iff exactly $h+1$ of the τ's rank u higher than v. The rest of the details (relationship between the Kemeny optimal permutation for the τ's and min. feedback edge set of G) are similar to the proof of Theorem 14, and are omitted.

For weighted tournaments, we may assume wlog. that all the edge weights $w(e)$ are odd. (If not, reduce the problem to the unweighted minimum feedback edge set problem via Cook's Theorem and Karp's NP-completeness result for the latter problem; then apply the trick from part (2) above, taking z to be an odd number.) As in the case of unweighted tournaments, start with an arbitrary list τ_1. For each edge $e=(u,v)\in E$, define $c(u,v)=w(u,v)-A(\tau_1,u,v)$, where $A(\tau_1,u,v)=1$ if τ_1 ranks u higher than v, and $A(\tau_1,u,v)=-1$ if τ_1 ranks v higher than u. Thus, $c(u,v)$ is the "deficit" between the weight of the edge (u,v) and the number of times the pair (u,v) have been ordered by τ_1 to be consistent with G. Note that $c(u,v)$ is even and non-negative. For every edge $e=(u,v)$ with $2\ell=c(u,v)\neq 0$, create 2ℓ lists $\tau_{e,i}=u,v,Z$; $\tau'_{e,i}=Z^R,u,v$; for $1\leq i\leq \ell$, where Z is an arbitrary permutation of $V-\{u,v\}$. Clearly, the total number of lists created is odd. The rest of the correctness proof is similar to the proof of Theorem 14, and omitted. □

B Rank Aggregation in Other Settings

B.1 Sports

The problem of determining the relative ranking of players or teams given the results of matches played between some of the pairs has received some attention from the mathematical community, see [26, 38, 37, 23, 1]. The general approach is to assume that each player i has an intrinsic strength s_i, and when two players meet the result of the match is a random variable whose parameters are determined by s_i and s_j. The ranking itself is determined as follows: first values for the strengths are chosen as those maximizing the probability of the given results and their sorted order defines the ranking. Two probabilistic models for the outcome as a function of the strengths are:

Bradley–Terry. When i and j with strengths s_i and s_j respectively are matched, player i wins with probability $\frac{s_i}{s_i+s_j}$ (this model is due originally to Zermelo [43]). Smith [37] justifies this model for games where the winner is the first to make n more "right" moves while its opponent makes a "wrong" move. This model is the basis for ranking in chess and tennis, see [23].

Mosteller. The player's true strength at any given match is a random variable r_i determined by s_i. When two players i and j are matched, i wins if $r_i > r_j$.

It is possible to use such methods for aggregating rankings: view every case where an element i is ranked above an element j as a match in which i has won and then apply one of the models above. However we are unsure of the applicability of this approach for two reasons: (1) there is no clear justification for the models in the search-engine setting, and (2) the resulting algorithms (for maximum likelihood) might be too complex. Nevertheless, it is worth trying to see how such methods actually perform.

B.2 Learning Theory

Rank aggregation has been studied in the machine learning literature in the context of meta-Search [10]. As in our work, the goal is to combine the output of several search engines. However, in contrast to our minimalist approach, previous work on meta-search involves *training* the meta-search function through the use of feedback. Ours is an appropriate model for meta-search in which the individual engines may be arbitrary subsets of a large set and/or may be modified at any time, without notice. In this setting there may be *no* opportunity for training before aggregation is required.

We note in passing that [10] also address (essentially) the problem of approximating minimum feedback edge sets, which is equivalent (see Section A) to approximating Kemeny optimal permutations. They present a detailed proof that a greedy algorithm gives a 2-approximation to the complement problem of finding a maximum acyclic subgraph; they overlook the fact that for any permutation π on the set of vertices, either π or reverse of π gives a 2-approximate solution to the maximum acyclic subgraph problem.

A second vein of related work within the machine learning community deals with collaborative filtering [32, 33, 34], where the goal is to create automated recommendation systems.

B.3 Database Middleware

Fagin [18] (see also [19]) considered the issue of determining an order among objects where each object has numerical values in for each of k fields. The assumption is that the objects are given in k lists where each list is sorted according to one of the fields. The ranking is done by an *aggregation function* t that acts on the k fields of an object and assigns such a combination some real value. Examples of aggregation functions are $\min$ and average. One interesting feature of this approach is that often it is not necessary to scan all the lists in order to determine the top m objects. This issue arises in several scenarios, including middleware for combining multimedia databases.

One can use such an approach for the problem of aggregating several orders by assigning for each position in each order some fixed value and then fixing some aggregation function. The Borda count as well as any other positional method are such examples.

C Two Proofs

Proof: (of Theorem 10) Let the alternatives be A, B, and C, and assume the majority graph contains the directed cycle (A, B, C). The function f must break the cycle and therefore will contradict an edge, say, $f(B) < f(A)$, meaning that f ranks B above A. Now remove the candidate C from the set of alternatives (and hence from $\tau_1, \ldots, \tau_k$). The majority prefers A to B, and there is no longer a cycle, so A is the Condorcet winner; however f selects B. $\square$

Proof: (of Lemma 13) Given $\tau_1, \tau_2, \ldots, \tau_k$ on n candidates, define a weighted complete bipartite graph $(C \cup P, W)$ as follows. $C = \{1, \ldots, n\}$ denotes the set of candidates (pages), $P = \{1, \ldots, n\}$ denotes the n available positions, and the cost $W(c, p)$ is the total footrule distance (from the τ_i's) of a permutation that places candidate c at position p, given by $W(c, p) = \sum_i |\tau_i(c) - p|$. It is easy to see that a permutation that minimizes the total footrule distance to the τ_i's is given by a min-cost perfect matching. An analagous argument is applied for the case of Spearman's correlation function.

Given the special nature of the cost function, it is conceivable that there are faster algorithms for finding min-cost perfect matchings on these graphs. $\square$