A Computational Study of the Kemeny Rule for Preference Aggregation

Andrew Davenport and **Jayant Kalagnanam**
IBM T.J.Watson Research Center, Yorktown Heights, New York, 10598
davenport@us.ibm.com, jayant@us.ibm.com

Abstract

We consider from a computational perspective the problem of how to aggregate the ranking preferences of a number of alternatives by a number of different voters into a single consensus ranking, following the majority voting rule. Social welfare functions for aggregating preferences in this way have been widely studied since the time of Condorcet (1785). One drawback of majority voting procedures when three or more alternatives are being ranked is the presence of cycles in the majority preference relation. The Kemeny order is a social welfare function which has been designed to tackle the presence of such cycles. However computing a Kemeny order is known to be NP-hard. We develop a greedy heuristic and an exact branch and bound procedure for computing Kemeny orders. We present results of a computational study on these procedures.

Introduction

Preference aggregation concerns how to combine the preference rankings of a number of alternatives by a number of different voters into a single consensus (or society) ranking. The aggregation of preferences in this way arises in applications such as determining the winners of elections or sports tournaments (Truchon 1998), multi-criteria and word association queries in databases (Dwork et al. 2001), the ranking of suppliers by buyers during strategic sourcing and the combining of search results from multiple search engines in order to fight spam (Dwork et al. 2001).

The properties of social welfare functions for aggregating preferences have been studied by mathematicians since the 18th century, and it is well known that complications and paradoxes can arise when there are more than two alternatives to be ranked. Arrow's impossibility theorem (Arrow 1951) states that no non-dictatorial social welfare function can ever satisfy a set of desirable and compelling properties for fair elections simultaneously on all domains of preferences. The debate on the merits of different types of social welfare functions is still ongoing (Saari and Valognes 1998), however two classes of ranking methods have been widely studied. Positional methods, such as the Borda count (Borda 1781), assign points to each alternative, depending on how they are ranked by each voter. Candidates are then ordered in the consensus ranking according to the sum of their assigned points. Majority ranking methods determine an outcome in terms of the majority ranking for the alternatives: alternative x is ranked ahead of alternative y in the consensus ranking if more voters prefer x to y. Positional methods such as the Borda count, while being very simple to compute, are considered to be highly manipulable and fail to satisfy important properties such as the independence of irrelevant alternatives (changes in individuals' rankings of "irrelevant" alternatives outside of a certain subset should have no impact on the societal ranking of this subset) and the Condorcet criterion (if some alternative is ranked ahead of all other alternatives by an absolute majority of voters, then it should be ranked first in the consensus ranking). Majority ranking methods such as Condorcet methods may be subject to cycles in the majority preference relation when there are more than two alternatives to be ranked, and thus may fail to select any winners at all (this is known as the Condorcet paradox (Condorcet 1785)).

The Kemeny rule (Kemeny 1959) has been proposed as a way of seeking a compromise ranking in the majority vote when there are cycles present in the majority preference relation. The Kemeny rule satsifies the Condorcet criterion and a weaker version of local independence of irrelevant alternatives. However the computational drawback of the Kemeny Rule is that is NP-hard to compute (Cohen, Schapire and Singer 1999; Dwork et al. 2001). This has discouraged the development of exact algorithms for computing Kemeny orders. Instead greedy heuristics (Cohen, Schapire and Singer 1999) or tractable multi-stage methods have been developed that combine both positional and majority voting methods (Black 1958; Dwork et al. 2001). A disadvantage of such approaches is that their theoretical properties are not known, and hence their outcomes can be unpredictable. Furthermore, NP-hardness is a only worst case complexity result which may not reflect the difficulty of solving problems which arise in practice. In the sections which follow we describe new exact and greedy heuristic procedures for computing Kemeny orders, and present results of an initial computational study into solving problems using these procedures.

Notation

Let X be a set of m alternatives and N be a set of n voters. Each voter j has weak order or ranking r^j of the alternatives in X. Each element r_j^s is the rank of voter j of alternative s. A ranking with no tie for a rank is an order on X, which can be represented as a sequence $(s_1,...,s_n)$,

where s_i is the alternative with rank i. A *preference aggregation function* (also known as a social welfare function) maps a set of rankings R (also referred to as a *profile of rankings*) into a single consensus ranking.

For each pair of alternatives s and t ranked by some profile of rankings R, we define v_{st} as the number of voters expressing a preference for s over t in their individual rankings, that is $v_{st}(R) = |\, j \in N : r_s^j < r_t^j \,|$. If $v_{st} > v_{ts}$ we refer to $s > t$ as the *majority vote* and $t > s$ as the *minority vote* for s and t. We represent a profile of rankings R by a weighted, directed graph we call the *preference graph*. Each alternative in X is represented by a node in the graph. Between each pair of nodes representing alternatives s and t is an edge of weight v_{st}.

We represent the strict majority relation by a simple, directed, weighted graph called the *strict majority graph*, which we also refer to more simply as the majority graph. There is an edge between each pair of nodes representing alternatives s and t iff $v_{st} > v_{ts}$. The weight of such an edge is $v_{st} - v_{ts}$. The weak majority relation is represented by the *weak majority graph*, which has an edge between each pair of nodes representing alternatives s and t iff $v_{st} \geq v_{ts}$. The weight of such an edge is $v_{st} - v_{ts}$.

Majority voting and the Condorcet criterion

The *Condorcet criterion* (Condorcet 1785) states that if some alternative is ranked ahead of all other alternatives by an absolute majority of voters, then it should be ranked first in the consensus ranking. When such an alternative exists, it is called the *Condorcet winner*.

Example 1: *Consider the preferences expressed by the profile (A,B,C), (A,C,B), (B,A,C) (e.g. the 1st voter prefers A to B to C, the 2nd voter prefers A to C to B, the 3rd voter prefers B to A to C). The preferences for each of the pairwise rankings are: $v_{AB}=2, v_{BA}=1, v_{BC}=2, v_{CB}=1, v_{AC}=3, v_{CA}=0$. We have a majority of 1 vote for $A > B$, a majority of 1 vote for $B > C$ and a majority of 3 votes for $A > C$.*

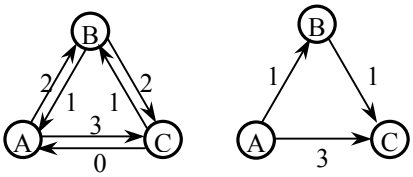

Figure 1: The preference graph (left) and majority graph (right) of the profile of Example 1

In Example 1 alternative A is the Condorcet winner. Note that when there is a Condorcet winner, it can be found by identifying the node in the weak majority graph with an indegree of zero. However, there may not always be a Condorcet winner, as cycles may occur in the majority ranking of alternatives.

Example 2: *Consider the profile of rankings (A,B,C), (B,C,A), (C,A,B). There is a majority of 1 vote for $A > B$, a majority of 1 vote for $B > C$ and a majority of 1 vote for $C > A$. This profile defines a cycle in the majority relation of $A > B$, $B > C$, $C > A$.*

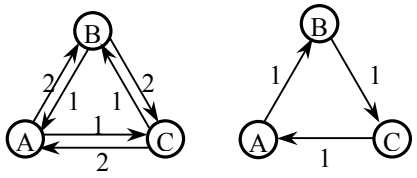

Figure 2: The preference graph (left) and majority graph (right) for the profile of Example 2

Cycles do occur in practice, even for quite small problems. For example, in a study of Olympic figure skating contests, cycles were found in competitions involving 9 out of 23 skaters (Truchon 1998). When cycles are present in the majority relation, the *extended Condorcet criterion* (Truchon 1998) gives a partial ordering of alternatives (with no cycles it gives a complete ordering). The extended Condorcet criterion states that if there is a partition X_C, X_D of X such that for any s in C and any t in D if the majority prefers s to t ($v_{st} > v_{ts}$) then s must be ranked ahead of t in the consensus ranking. One application of the Condorcet and extended Condorcet criteria is in fighting ``spam'' in individual rankings (Dwork *et al.* 2001). However it has been shown that no positional method for aggregating preferences can ever satisfy either of these criteria (Young 1974).

Given some partial or complete consensus ranking of a profile of rankings R, we can enforce the extended Condorcet criterion on the ranking in the following way[1]:

Algorithm 1:

1. Identify all the maximal strongly connected components in the majority graph of R. Each strongly connected component of size > 1 node identifies a set of nodes which are involved in at least 1 cycle.
2. For each node n in each component, consider all target nodes m on all outgoing edges of n. If m is not in the same component as n, add to the consensus ranking the ordering $s > t$, where s is the alternative represented by node n and t is the alternative represented by node m.

Alternatives represented by nodes within non-trivial strongly connected components of the majority graph are involved in cycles, and will not be ordered by Algorithm 1. Kemeny (Kemeny 1959) proposed a way of breaking such cycles, using a notion of distance for orders. Given a total order r, a weak order r^j and 2 alternatives s and t, we define:

$$\delta_{st}(r, r^j) = \begin{cases} 1 & \text{if } r_s < r_t \text{ and } r_t^j \leq r_s^j \\ 0 & \text{otherwise} \end{cases}$$

and

$$\Delta(r, r^j) = \sum_{s \in X} \sum_{t \in X} \delta_{st}(r, r^j)$$

[1] A similar scheme is presented in (Cohen, Schapire and Singer 1999) and in (Dwork et al. 2001).

The value of $\delta_{st}(r, r^j)$ indicates whether there is a disagreement in the relative ranking of s and t between r and r^j. $\Delta(r, r^j)$ measures the total number of disagreements between r and r^j, and is referred to as the *Kendall tau distance* (Diaconis 1988). For complete orders, the Kendall tau distance is the ``bubble sort'' distance: it gives the number of pairwise adjacent transpositions to transform one order into another. The distance between a complete order r and a profile of rankings R is given by $d(r, R) = \sum_{j=1..n} \Delta(r, r^j)$. A *Kemeny order* for a profile R is an order r which minimizes the distance $d(r, R)$, i.e. an order which has the minimum number of disagreements with the pairwise rankings in the profile R. It has been shown that any Kemeny order satisfies the extended Condorcet criterion (Truchon 1998).

Algorithm design

Problem formulation

Given a (weak) majority graph G_R of a profile of rankings R, we wish to compute a total order of the alternatives in R taking into account as many of the majority preferences expressed by the edges in G_R as possible. If G_R is acyclic, a topological sort of the nodes in the graph will identify a Kemeny order of the alternatives. When G_R contains cycles, it becomes necessary to remove some set of edges from G_R such that the resulting graph is acyclic. To find a Kemeny order, the sum of the weights of the edges removed from the graph must be minimized. This problem is known as the minimum weight feedback edge set problem (Festa, Pardalos, and Resende 1999), and is known to be NP-hard. Feedback set problems have received considerable attention in recent years (Festa, Pardalos, and Resende 1999), especially with respect to approximation algorithms and tractable cases. However, as far as we are aware, there has been no work on exact approaches for finding optimal solutions to the weighted variant of the feedback edge set problem in directed graphs. In the sections which follow, we outline a new greedy heuristic and exact branch and bound approach for finding Kemeny orders, based on formulating the problem as minimum weight feedback edge set problem.

Solution representation

We use a *solution graph* to represent orderings of alternatives that are established during a search procedure when constructing a Kemeny order. Each alternative in the set X is represented by a node in the solution graph. A directed edge between a pair of nodes representing alteratives s and t indicates that in the solution s is ranked ahead of t. A solution graph which is simple, directed, acyclic and where all nodes have degree of $|X|-1$ represents a complete ordering of the alternatives in X. The goal of the search procedure is to construct a solution graph representing a complete ordering which is a Kemeny order. A complete ordering can be obtained from the solution graph by performing a topological sort of the nodes in the graph.

We often want to compute a majority graph which takes into account the orderings that have been made in a solution graph, even if some of these orderings correspond to minority votes. Thus all edges in the solution graph have corresponding edges in the majority graph representing the same orderings, regardless of the majority votes for these alternatives. In the case where an edge in the solution graph corresponds to an ordering with a minority vote, the reverse edge in the direction of the majority vote will not be present in this modified majority graph.

Greedy heuristic and branch and bound search

We have developed a simple greedy heuristic and an exact depth-first branch and bound procedure for computing Kemeny orders, based on constructing a solution by identifying orderings of pairs of alternatives and adding the corresponding edges to the solution graph. We use the following heuristic to select and order a pair of alternatives at each node of the search in both of these approaches: select the pair of alternatives which has the greatest difference between the majority and the minority vote for the different pairwise orderings and order the alternatives following the majority vote.

The greedy procedure simply follows this heuristic and applies propagation rules after each ordering is made, terminating when the solution graph represents a complete order. The branch and bound procedure performs a depth-first backtracking search, following the heuristic at each node and applying lower bounding procedures in addition to the propagation rules to prune the search. In the sections which follow we discuss the propagation rules and lower bounding techniques.

Propagation

A solution is built up incrementally by adding edges to the solution graph. In order to ensure that the solution graph remains acyclic we maintain transitive closure of the graph after edges are added or deleted. Since the graph is directed and acyclic, we can use an efficient incremental algorithm that maintains transitive closure in time $O(n^2)$ for n nodes in the graph.

The first step for both approaches is to run Algorithm 1 to satisfy the extended Condorcet criterion of the ordering. If there are no cycles in the majority graph, this step is sufficient to identify a Kemeny order. Otherwise, as edges are added to the solution graph during search, cycles may be broken in the majority graph that takes into account solution graph orderings (and new cycles may be created). We run Algorithm 1 as a propagation step on this majority graph. If there are new strongly connected components in the majority graph as a result of edges added to the solution graph, further orderings of alternatives may be added to the solution. Applying this propagation step every time an edge is added to the solution graph ensures that we only

ever branch on orderings that are involved in some cycle in the majority graph associated with some solution graph.

Lower bounding

A simple lower bound LB_1 on the Kemeny distance of a Kemeny order is the sum of the minority votes for every pair of alternatives:

$$LB_1 = \sum_{\forall s \in X} \sum_{\forall t \in X, s \neq t} \min(v_{st}, v_{ts})$$

This bound will be equal to the Kemeny distance of the Kemeny order when there are no cycles in the majority graph: in this case we can follow the majority vote in the Kemeny order for the ranking of all pairs of alternatives. When there are cycles, the bound can be strengthened during search by taking into account the orderings of the pairs of alternatives that have been fixed by the search procedure.

A stronger lower bound can be computed by determining a set of edge disjoint cycles in each strongly connected component of the majority graph (associated with some solution graph). Each edge disjoint cycle contains at least one edge representing a majority vote between a pair of alternatives which the Kemeny order must disagree with. The impact of disagreeing with the majority vote on the distance of the Kemeny order for some pair of alternatives s and t is $v_{st} - v_{ts}$, which is the weight of the edge between the nodes corresponding to the pair of alternatives in the majority graph. We can add, for each edge disjoint cycle, the minimum positive weight of the edges in the cycle to the bound LB_1 to form a stronger lower bound.

We simplify the lower bound computation by considering edge disjoint cycles containing only 3 nodes. This is motivated by the observations that (a) it is harder to compute a Kemeny order for profiles where there are no ties in the rankings between any pairs of alternatives[2] and (b) in a majority graph where there are no ties, there exists an edge between every pair of nodes in every non-trivial strongly connected component. When this is the case then every set of nodes involved in a cycle of length greater than 3 must contain a cycle of exactly length 3 on a subset of these nodes. An illustration of this proposition is given in Figure 3. Consider a cycle of length 4 between nodes A, B, C and D. When there are no ties, there must be an edge in the majority graph between nodes B and D. If this edge is directed from node B to D then we have the 3-cycle involving nodes A, B, D (left, Figure 3). If the edge is oriented the reverse way, we have the 3-cycle involving nodes B, C, D (middle, Figure 3). Finally, we consider the case where a cycle contains more than 4 nodes. The right graph in Figure 3 shows a cycle involving 5 nodes A, B, C, D, E. Consider the edge between nodes A and D which is not an edge of the 5-cycle. If this edge is directed from nodes A to D then we have the 3-cycle involving nodes A, D, E. Otherwise, the direction from D to A forms a 4-cycle involving nodes A, B, C, D for which we have shown there must exist a 3-cycle on a subset of these nodes. A similar argument can be used to show that each cycle involving n nodes ($n>3$) must contain a cycle involving either 3 or $n-1$ nodes.

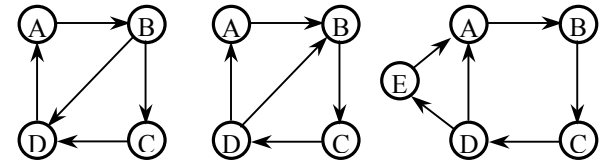

Figure 3: Illustration of 3-cycles in a majority graph.

We formulate the problem of computing edge disjoint 3-cycles in a strongly connected component of the majority graph as an iterative min-cost max-flow problem (Ahuja, Magnanti, and Orlin 1993). An example of the network formulation we use to solve this problem is given in Figure 4 for the majority graph illustrated in the left of Figure 3.

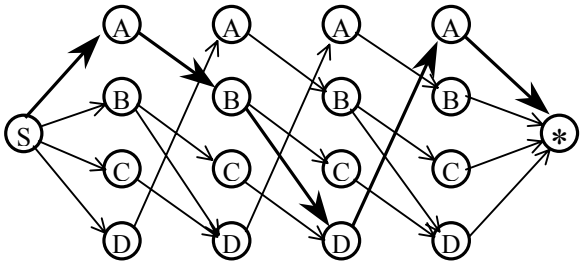

Figure 4: An example network formulation for the majority graph in the left of Figure 3.

The network has four layers of nodes, $l_1,...,l_4$. In each layer there are $|X|$ nodes, where each node corresponds to a node in the majority graph. There is a directed edge between each pair of nodes in layers l_i and l_{i+1} in this network if there is a directed edge between the corresponding pair of nodes in the majority graph. The cost on each edge between the layers is some constant K (e.g. the number of voters) minus the weight between the corresponding pair of nodes in the majority graph (we want to find cycles with the largest possible minimum edge cost in the majority graph). All nodes in layer l_1 are connected to the source node and all nodes in layer l_4 are connected to the sink node of the network. These connections all have a cost of 1. All minimum edge capacities are set to 0. The maximum capacities of all the edges between nodes in the layers are to set to 1. During each iteration we solve a min-cost max-flow problem to determine a set of paths through the network starting and ending at some common node in layers l_1 and l_4, representing a starting and ending point for a 3-cycle in the majority graph. For example, the highlighted edges

[2] This situation will always occur when we have an odd number of voters and all voters rank all alternatives.

in Figure 4 illustrate a path of length 3 starting at node A in layer l_1 and ending at node A in layer l_4. Such a path corresponds to the 3-cycle *(A, B, D, A)* in the majority graph in the left of Figure 3. In each iteration we select and fix the common starting and ending nodes n_s and n_e in layers l_1 and l_4 (e.g. node A in Figure 4). The maximum number of 3-cycles max_3 that can be found from these nodes is the minimum of the out-degree of node n_s and the in-degree of node n_e. We set the maximum capacity of the edge from the source node to the node n_s and node n_e to the sink node to max_3. We set the maximum capacities of all remaining edges from the source node to the layer l_1 nodes to 0. Similarly, we set the maximum capacities of all remaining edges between nodes in layer l_4 and the sink node to 0. The solution to the min-cost max-flow problem on this network specifies a set of paths through the network that correspond to a set of 3-cycles in the majority graph starting and ending at the fixed node. The value of the flow is the number of disjoint 3-cycles found in the corresponding majority graph. We determine, for each cycle found by the network flow solution, the lowest cost of all the edges in the cycle and add (the cost constant K minus) this lowest cost to LB_1. Enumerating the paths can be achieved by solving multiple shortest path problems, one for each path in the network flow solution. Each shortest path computation can be performed with time complexity $O(n+m)$ for a directed acyclic graph with n nodes and m edges. Finally, at the end of each iteration, we delete all edges from the network which were found in the min-cost max-flow solution, to ensure that a single edge is only assigned to one 3-cycle. We have also observed that the order in which we perform the network flow iterations can have an impact on the number of 3-cycles that are found. For each node in the majority graph we determine the minimum and maximum of their in-degree and out degree. We sort the nodes by decreasing minimum of these degrees, breaking ties by decreasing maximum of these degrees, and perform the network flow iterations in this order.

The time to compute the min-cost max-flow solution of the network dominates the other parts of the procedure. This can be computed in time $O(m \log U(m + n \log n))$ for n nodes of maximum degree U and m edges (Ahuja, Magnanti, and Orlin 1993). Since we may have up to n iterations, the time complexity of this lower bounding procedure is $O(nm \log U(m + n \log n))$.

Experiments

Problem generation

We evaluated the algorithms on randomly generated problems. For each problem, we first generated a total order representing a consensus ordering of m alternatives. We then generated a preference graph where each one of n voters agrees with the consensus ordering regarding the ranking of every pair of alternatives with some consensus probability p. We generate problems with an odd number of voters, where each voter ranks all alternatives with no ties.

Results

The first results we present explore the effect of varying the consensus probability and the number of voters on problem solving difficulty[3]. CPU time results for 15 alternatives are presented in Figure 5 (for all experiments, 500 problems are solved at each data point). As we increase the consensus probability, the CPU time required to find a Kemeny order decreases dramatically. When the consensus probability is over 0.7, most problems are solved without any search. Increasing the number of voters also makes problems easier to solve at higher consensus probabilities, but harder to solve at lower probabilities. We do not have an explanation for the phenomena at low probabilities. Figure 6 shows the difference in quality between the solution found by the greedy heuristic procedure and optimal solution found by branch and bound for the same set of problems. As we increase the number of voters and the consensus probability, the greedy heuristic finds better solutions. We have seen similar results for larger problems.

The encouraging aspect of these results is that although finding a Kemeny order can be very difficult for problems with very little consensus, it becomes much easier for problems with a reasonable amount of consensus that we would hope to find in real applications. One question which arises is: what are the range of problems we can solve to optimality by generating solutions with a greedy algorithm and comparing their Kemeny distance to the min-cost max-flow Kemeny distance lower bound? We compare here our greedy heuristic procedure with the greedy heuristic presented in (Cohen, Schapire, and Singer 1999) which has an approximation factor of 2[4]. Figures 7 and 8 present results exploring the deviation between the min-cost max-flow lower bound and the greedy heuristic solutions for finding Kemeny orders for problems with 50 alternatives. For 25 voters, we can prove optimality of most problems with consensus probability greater than 0.7 by comparing the solution of our greedy heuristic procedure with the lower bound. With only 5 voters, this consensus probability increases to around 0.9. The greedy procedure of Cohen et al. does less well at high probabilities, but finds better solutions at low probabilities. We have also seen similar results for problems with different numbers of alternatives.

[3] All experiments were run on a 1GHz Pentium III computer using C++ implementations of the algorithms

[4] The problem studied in (Cohen, Schapire, and Singer 1999) also allows voters to express degrees of preferences between alternatives.

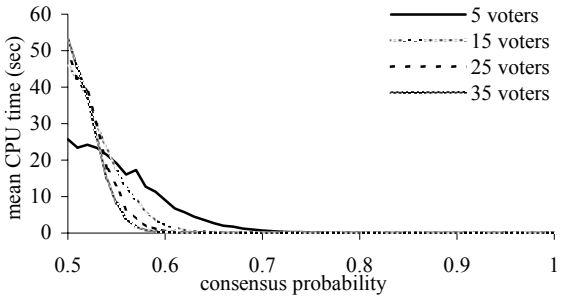

Figure 5: Mean CPU time to find optimal solutions for problems with 15 alternatives, using branch and bound.

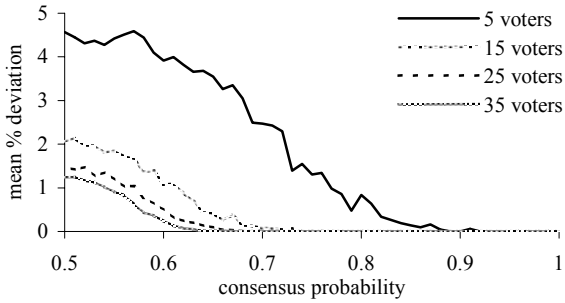

Figure 6: Mean % deviation of greedy heuristic procedure solutions from optimal, 15 alternatives.

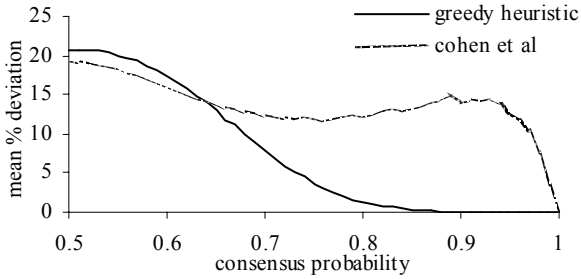

Figure 7: Mean % deviation of greedy solutions from lower bound, 50 alternatives, 5 voters.

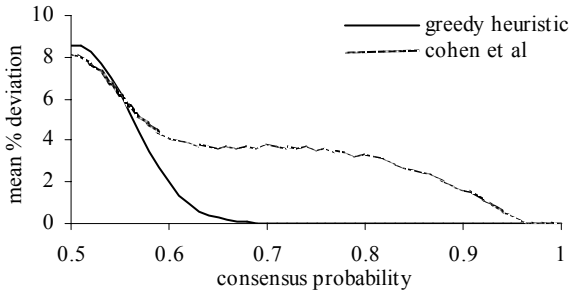

Figure 8: Mean % deviation of greedy solutions from lower bound, 50 alternatives, 25 voters.

Conclusions

We have presented new exact and heuristic algorithms for aggregating preferences following the Kemeny rule. Results of a computational study indicate that computing Kemeny orders appears to be computationally expensive when there is very little consensus between voters. However we have found that when there is a reasonable degree of consensus, a Kemeny order can be found within a short amount of time. Furthermore, compared to multi-stage techniques that combine both positional and majority voting methods, the Kemeny orders found by these algorithms have well studied theoretical properties. Further research into improving lower bounding techniques may significantly extend the range of problems for which Kemeny orders can be found.

References

Ahuja, K.; Magnanti, T.; and Orlin, J. 1993. *Network flows.* Prentice Hall.ISBN 0-13-617549-X.

Arrow, K. 1951. *Social Choice and Individual Values.* John Wiley, New York.

Black, D. 1958. *Theory of Committees and Elections.* Cambridge University Press.

Borda. 1781. Mémoire sur les élections au scrutin. In *Histoire de l'Académie Royale des Sciences*.

Cohen, W.; Schapire, R.; and Singer, Y. 1999. Learning to order things. *Journal of Artificial Intelligence Research* 10:213-270.

Condorcet. 1785. Essai sur l'application de l'analyse à la probabilité des décisions rendue à la pluralité des voix. In *Paris: Imprimerie royale*. Also reproduced in Condorcet, Sur les élections et autres textes, edited by O. de Bernon, Fayard, 1986.

Diaconis, P. 1988. *Group representation in probability and statistics.* IMS Lecture Series 11, Institute of Mathematical Statistics.

Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. 2001. Rank aggregation methods for the web. In *Proc. 10th WWW*, 613-622.

Festa, P.; Pardalos, P.; and Resende, M. 1999. Feedback set problems. In *Handbook of Combinatorial Optimization,* vol. 4. Kluwer Academic Publishers.

Kemeny, J. 1959.Mathematics without numbers. *Daedalus* 88:571-591.

Saari, D., and Valognes, F. 1998. Geometry, voting, and paradoxes.*Mathematics Magazine* 71(4):243-259.

Truchon, M. 1998. Figure skating and the theory of social choice. Technical Report Cahier 98-16, Centre de Recherche en Economie et Finance Appliquées, Université Laval, Canada.

Young, H. 1974. An axiomitization of Borda's rule. *Journal of Economic Theory* 9:43-52.