A Decision-theoretic Model of Rank Aggregation

Tyler Lu and Craig Boutilier
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
{tl, cebly}@cs.toronto.edu

Abstract

Modern social choice theory has spurred considerable recent work in computational rank aggregation, the problem of aggregating rankings into a consensus ranking. However, in the social choice context where rankings represent preferences of agents, or “voters,” over outcomes, or “candidates,” there is scant justification in the literature for producing rankings as an output of the aggregation process. We introduce a novel model which offers one possible decision-theoretic motivation for constructing a consensus ranking rather than an outcome. Roughly speaking, we view rankings as a compact, easily communicable decision policy in the face of uncertainty about candidate availability. This gives rise to a principled approach, based on minimizing expected dissatisfaction with the candidate choice, for optimization functions on rankings. We provide exact and approximation algorithms for computing optimal rankings in our model that exploit the properties of our objective function. We also describe interesting connections to popular voting protocols such as the plurality rule and the Kemeny consensus.

1 Introduction

Social choice theory is concerned with the problem of aggregating preferences of individual agents over some decision or outcome space to determine a suitable consensus outcome. Arrow [1] famously considered the problem of aggregating rankings of outcomes (or candidates) by agents (or voters) not into a choice, but into a consensus ranking. Since this time, the problem of rank aggregation has attracted considerable attention in social choice. Allowing voters to express rankings rather than single votes has clear value: the relative preference of a voter for any candidate—not just its top-ranked—can sensibly play a role in determining the winner, and does in common rules like Borda. By contrast, much work in social choice assumes that the output of an aggregation procedure should itself be a ranking of all candidates. A variety of models take just this approach, e.g., the Kemeny consensus, which produces a ranking that minimizes the sum of pairwise candidate “disagreements” between the votes (input rankings) and the result. However, within the context of social choice problems such as voting the need to produce a consensus ranking rather than a consensus outcome is often left unspecified and unjustified.

Consider the popular Kemeny consensus: if the goal is to produce a single winner, why should one produce a ranking? And if this ranking is used to produce a winner, why should pairwise disagreements across the entire ranking of each voter be minimized? Of course, there are a variety of rank aggregation settings where the decision space explicitly requires a ranking. For example, if each “voter” is expressing a noisy assessment of some underlying objective ranking (e.g. quality of sports teams), under certain assumptions, the Kemeny consensus provides a maximum likelihood estimate of the underlying ranking [10]. In web search, one might want to rank results to minimize average effort to find the relevant results [2, 4, 5]; but it is unclear why social choice/voting models are appropriate. Instead, a model that explicitly models search costs and probability of relevance would be more appropriate [9].
More generally, we argue that the decision criterion for which aggregation is being implemented should directly influence the process by which one aggregates rankings. To this end, we propose a new model that justifies the use of rankings in preference aggregation. Our model supposes that any candidate may be rendered unavailable with a certain probability, and the output ranking determines a decision (or “winner”) by selecting the best available candidate. In our model, the optimal aggregate ranking minimizes the expected voter “dissatisfaction” with the selected candidate. To illustrate, suppose an organization considers a number of candidates for an open position. Members of the hiring committee submit their preference orderings; however, there is a chance any candidate may take another job, so the committee must be prepared to select a candidate from any subset of available candidates. A ranking—where the top-ranked available candidate is chosen—provides us with a compact, easily interpretable policy for selection.

In this short abstract, we briefly develop this model formally and sketch some theoretical and computational results, including the relation of this new aggregation model with the Kemeny consensus. We also discuss some directions for future research, including how this model can be used to support personalization in a learning setting, and suggesting alternative, decision-theoretically motivated approaches to rank aggregation.

Considerable work on computational social choice has focused on the Kemeny aggregation rule [7]. It has been shown to be NP-hard to compute [6, 4], but good approximation heuristics have been given in the context of web meta-search [4] as well as a polynomial time approximation scheme (PTAS) [8]. Practical approaches for exact computation have also been explored [3]. While the Kemeny rule is quite popular and natural, it does not address why one needs an aggregate ranking. Our model gives intuitions as to why a Kemeny consensus can be useful from a decision-making perspective.

Rank or preference aggregation also has some interesting connections to the literature on rank learning. For example, [2] focuses on learning a preference function over all pairs of items while only given preference comparisons between some of the items. A final ranking of all items that is most consistent with the learned preference function is computed. This can be seen as preference aggregation where the preference comparisons come from different users and the learned ranking is an aggregation of these preferences. In settings where we have little information on each user’s preferences over items, as in recommender systems, the preferences of similar users can be aggregated under the assumption that similar users have similar preference functions. Thus we can leverage the more abundant aggregated preferences to facilitate better learning. This is roughly the idea behind, for example, the “label ranking paradigm” of [5] where one is interested in building personalized ranking for each user while only having limited preference data.

2 Model and Technical Results

We have voters $N = \{1, 2, \ldots, n\}$ with corresponding total preference orderings $V = \{v_1, \ldots, v_n\}$ (i.e. votes or rankings) over the set of candidates $C = \{c_1, \ldots, c_m\}$. Denote by $v_\ell(c_i)$ the rank of $c_i$ in vote $v_\ell$ so that voter $\ell$ prefers $c_i$ over $c_j$ if $v_\ell(c_i) < v_\ell(c_j)$. From a decision-theoretic perspective, we are interested in aggregating voters’ orderings to select one candidate (i.e. decision). However, in some real life scenarios we cannot expect any particular candidate to be available. Some decisions turn out to be unavailable and backups must be planned in advance. One solution to this problem is to provide a complete ranking of all the candidates, and upon knowing which candidates are available, the top available candidate of the ranking is selected. Thus, ranking here is useful in the sense that it acts as a policy of what to do under uncertain outcomes. The next question is how do we rank the candidates so as to maximize the voters’ “happiness”? But first we give a formalization of the above discussion.

**Definition 1.** A decision policy is any function $W : 2^C \rightarrow C \cup \{\bot\}$. That is, given the set of available candidates, $W$ outputs a candidate to recommend. We further restrict that $W(\emptyset) = \bot$, that is, the decision policy does not make a recommendation when no candidates are available.

Let $W$ be a class of decision policies and $P$ a probability distribution over $2^C$ giving the probability that exactly a certain subset $S \subseteq C$ is available. For a ranking $r$ and available candidates $S \subseteq C$, let $\text{top}(r, S)$ be the highest ranked candidate in $r$ that also occurs in $S$. Our goal, given input $V$ and $P$,
Theorem 2. Let just the top candidates. In fact, we can show that our model really becomes a Kemeny aggregation bottom matters a lot too. Thus it becomes important to get nearly all of the ranking correct and not p the top of rankings. When set of candidates p voting. First it can be seen that when There are several interesting connections between our model, Kemeny aggregation, and plurality voting. Algorithmically, one can compute the optimal ranking using an integer programming formulation. The basic idea is to find the myopically dominant candidate has an approximation ratio of. We have also developed a very natural greedy heuristic to approximate the optimal solution. This is repeated for all m positions. The heuristic can be improved as follows: we detect if there’s a dominant candidate or Condorcet winner at each stage: a candidate having more than half of top votes (w.r.t. the remaining candidates) and choose it first (it can be shown this is optimal). Although we have only very loose bounds on the approximation ratio, our experiments suggest that this heuristic often finds the optimal solution, and is very close to optimal otherwise. We also have a PTAS guaranteeing arbitrarily good approximations, which can be implemented by modifying our integer programming formulation of exact computation. The basic idea is to find the myopically top k candidates, then arbitrarily order the remaining candidates. This offers a good approximation since the objective function penalizes “mistakes” much less near the bottom.

We denote the expected number of mistakes with p representing a candidate is unavailable. Suppose r = r1r2...rm, let \( W_p^{(1,k)}(r, V) := \sum_{i=1}^{m} \sum_{j=1}^{m} (1 - p) p^{r(r_i)} - (1 - p) p^{r(r_i) - t(r_i,r,u,v)} \). We conjecture that optimizing \( M_p(r, V) \) is NP-hard for values of p that are not too small. In fact, we will see later that, for values of p close to 1, an optimal ranking is also Kemeny optimal, implying NP-hardness. One nice property of the objective function is that, roughly, it penalizes exponentially less for misorderings near the bottom.

Algorithmically, one can compute the optimal ranking using an integer greedy heuristic to approximate the optimal solution. The idea is that we first find the top-ranked candidate r1 by greedily choosing the candidate with the least expected number of mistakes if it were placed in the top position (assuming that if it is unavailable, no decision is made). For the second position, we consider the remaining candidates and compute an updated expected number of mistakes that takes into account the fact that r1 is in the first position. This is repeated for all m positions. The heuristic can be improved as follows: we detect if there’s a dominant candidate or Condorcet winner at each stage: a candidate having more than half of top votes (w.r.t. the remaining candidates) and choose it first (it can be shown this is optimal). Although we have only very loose bounds on the approximation ratio, our experiments suggest that this heuristic often finds the optimal solution, and is very close to optimal otherwise. We also have a PTAS guaranteeing arbitrarily good approximations, which can be implemented by modifying our integer programming formulation of exact computation. The basic idea is to find the myopically top k candidates, then arbitrarily order the remaining candidates. This offers a good approximation since the objective function penalizes “mistakes” much less near the bottom.

Theorem 1. The algorithm that myopically selects the top k candidates, i.e. optimizes \( M_p^{(1,k)}(r, V) \), (and also considers dominant candidates) has an approximation ratio of \( 2p^k/(1 - p)^2 \).

There are several interesting connections between our model, Kemeny aggregation, and plurality voting. First it can be seen that when p = 0 the top candidate in the optimal ranking is also the one plurality voting would elect. This makes sense as our model would only focus the mistakes at the top of rankings. When p is very close to 1, our model tells us that the candidates ranked at the bottom matters a lot too. Thus it becomes important to get nearly all of the ranking correct and not just the top candidates. In fact, we can show that our model really becomes a Kemeny aggregation problem (making the optimization NP-hard).

Theorem 2. Let \( \epsilon = 2/(\min(m - 1, 2) \) and p > \( (1 - \epsilon)^{-\epsilon} \) then the following holds (1) for any set of candidates C, votes V, any minimizer \( r^* \) of \( M_p(r, V) \) is also a Kemeny optimal ranking, (2) there exists candidates C’, votes V’, and a Kemeny optimal ranking K’ which is not a minimizer of
\( M_p(\cdot, V') \). In fact this is true for all \( p \in (0, 1) \), and (3) for any set of candidates \( C \), votes \( V \), any Kemeny optimal ranking \( K^* \) we have \( M_p(K^*, V)/\min_r M_p(r, V) \leq 1/(1 - \epsilon) \).

A related question is how well the Kemeny optimal ranking compares in our mistake model. It turns out when \( p \) is small it is twice as bad.

**Theorem 3.** For any votes \( V \), let \( K^* \) be the Kemeny optimal ranking, then
\[
M_p(K^*, V)/\min_r M_p(r, V) \leq 2/(1 - p)^2.
\]
Furthermore, for small values of \( p \) there exists set of votes such that this ratio matches the upper bound very closely.

When \( p \) is larger the bound above is loose (e.g. if \( p \) is close to 1 we know Kemeny is a very good approximation of our optimization model). The basic idea is that Kemeny will always choose a candidate at the top if it has more than half of all top votes (coinciding with optimal ranking for small \( p \)) then the remaining ordering is not as important.

## 3 Conclusion and Future Work

We have introduced a novel model that provides one possible rationale for computing a consensus ranking of voter preferences rather than a decision. Our model induces a principled objective function for rank aggregation, that differs from classical rank aggregation rules, but bears some strong connection to methods such as plurality and Kemeny voting rules. We have provided exact algorithms as well as approximation algorithms that can exploit the fact that items higher in the ranking are more important.

Future work includes looking at other interesting and realistic distributions over available candidates, as well as dealing with aggregating partial preferences from voters. We are also exploring other decision models that would naturally induce rankings, such as web search and or other consensus recommendations that provide a range of options for a variety of users. One of our primary aims is to incorporate such decision models into techniques for rank learning where limited preference data from users must be aggregated to facilitate learning. In this setting, the tradeoff between making fully personalized decisions (with limited data) and pure consensus decisions (with increased degree of dissatisfaction) gives rise to natural criteria for clustering/aggregating certain subsets of user and not others. From a more technical perspective the complexity of our optimization problem remains open, as does proving tighter approximation bounds for the greedy algorithm.

## References


