The Unavailable Candidate Model: A Decision-Theoretic View of Social Choice

Tyler Lu
Department of Computer Science
University of Toronto, CANADA
tl@cs.toronto.edu

Craig Boutilier
Department of Computer Science
University of Toronto, CANADA
cebly@cs.toronto.edu

ABSTRACT

One of the fundamental problems in the theory of social choice is aggregating the rankings of a set of agents (or voters) into a consensus ranking. *Rank aggregation* has found application in a variety of computational contexts. However, the goal of constructing a consensus ranking rather than, say, a single *outcome* (or winner) is often left unjustified, calling into question the suitability of classical rank aggregation methods. We introduce a novel model which offers a decision-theoretic motivation for constructing a consensus ranking. Our *unavailable candidate model* assumes that a consensus choice must be made, but that candidates may become unavailable after voters express their preferences. Roughly speaking, a consensus ranking serves as a compact, easily communicable representation of a *decision policy* that can be used to make choices in the face of uncertain candidate availability. We use this model to define a principled aggregation method that minimizes expected voter dissatisfaction with the chosen candidate. We give exact and approximation algorithms for computing optimal rankings and provide computational evidence for the effectiveness of a simple greedy scheme. We also describe strong connections to popular voting protocols such as the plurality rule and the Kemeny consensus, showing specifically that Kemeny produces optimal rankings in the unavailable candidate model under certain conditions.

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Theory

Keywords

social choice, rank aggregation, voting, preferences

EC'10, June 7–11, 2010, Cambridge, Massachusetts, USA.
Copyright 2010 ACM 978-1-60558-822-3/10/06 ...$10.00.

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 [3] famously considered the problem of aggregating rankings of outcomes (or candidates) by agents (or voters) to produce a consensus ranking rather than a single choice (or winner). Since this time, the problem of *rank aggregation* has attracted considerable attention in social choice and popular aggregation schemes like the Kemeny consensus [15] have found wide application in computational contexts (e.g., web ranking). Rank aggregation methods have a key advantage over simple voting schemes like plurality. By taking rankings as *input* rather than single votes, they allow voters to express their relative preferences for candidates, which can arguably play a reasonable role in determining a winner (e.g., the Borda count does just this). However, the need to produce a consensus ranking as *output* of a social choice scheme is typically less well-motivated. Despite this, a variety of models take this approach, for instance, the Kemeny consensus, which produces a ranking that minimizes the sum of pairwise candidate disagreements between the input rankings and the output ranking. In a social choice context, the need to produce a consensus ranking rather than an 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? If this ranking is used merely 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 [25, 20]. Of course, this justification is applies more to settings where some underlying "true" ranking exists than to those in which genuinely distinct preferences are to be aggregated. In other circumstances, we might wish to find not a single winner, but a slate of k candidates for k positions. We could imagine using the top k candidates produced by some consensus ranking process, but existing rank aggregation schemes (e.g., Kemeny) may not be appropriate for this problem. In web search, one might want to rank results to minimize average effort to find the relevant results [5, 10, 14]; but again it is unclear why social choice models are suitable. Instead, a model that explicitly considers search costs and probability of relevance would be more appropriate [22].

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 motivates the output of a consensus ranking in preference aggregation. Our *unavailable candidate 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, consider an organization interviewing 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 accept a different 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 paper, we develop this model formally. Our model gives rise to an objective function over rankings that has some nice properties. Computationally, we conjecture the optimization of the consensus ranking to be NP-hard and give an integer programming (IP) formulation of the ranking optimization problem. We also provide a polytime approximation scheme (PTAS) as well as more practical greedy heuristics that exploit the structure and properties of our objective function; empirical results suggest that the greedy methods provide excellent approximations. We also show interesting connections to the Kemeny and plurality voting rules. For example, we show that an optimal ranking corresponds to a Kemeny consensus under certain assumptions, but that in general a Kemeny consensus can produce twice as many expected "disagreements" as the optimal ranking. Nevertheless, our model gives intuitions as to why a Kemeny consensus can be useful from a decision-theoretic perspective at least in certain cases. 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.

We emphasize that the unavailable candidate model is just one of a number of decision models that justifies the output of consensus rankings. Others, corresponding to the some of the applications mentioned above, will give rise to different aggregation algorithms and analysis, in some case, justifying the use of classical social choice schemes, but in others, requiring new preference aggregation methods.

2. PRELIMINARIES

We introduce the basic social choice setup and discuss relevant background concepts before presenting our model in the next section (for further background, see [12, 24]). We assume a set of *candidates* $C = \{c_1, \dots, c_m\}$ and *voters* $N = \{1, 2, \dots, n\}$. Let Γ_C be the set of bijections of the form $r : C \to \{1, \dots, m\}$, i.e., the set of permutations or *rankings* of C, mapping candidates to rank positions. For convenience, we often write a ranking r as a sequence, e.g., $r = bcad$ meaning $r(b) = 1, r(c) = 2$, etc.

Candidates can represent any outcome space over which the voters have preferences (e.g., political candidates, restaurants, building designs, public projects, etc.) and for which a single collective choice must be made. Voter ℓ's preferences are represented by ranking $v_\ell \in \Gamma_C$, where *ℓ prefers c_i to c_j* iff $v_\ell(c_i) < v_\ell(c_j)$. We refer to voter rankings as *votes*, and a collection of votes $V = (v_1, ..., v_n) \in \Gamma_C^n$ as a *preference profile*. Abusing notation, we write $v \setminus S$ to refer to the restriction of a vote or ranking v to candidates *not* in $S \subseteq C$ (similarly for $V \setminus S$). Our aim is to choose a candidate or *winner* from C that implements some social choice function $f : \Gamma_C^n \to C$, where f reflects some social desiderata (e.g., maximizing happiness, fairness, etc.). In many social choice models, however, a *rank aggregation function* $f : \Gamma_C^n \to \Gamma_C$ is used which aggregates preference profile V into a consensus ranking $f(V)$. This ranking can be used to produce a winner by taking the top-ranked candidate.

For winner determination, *plurality* is the simplest and most popular scheme: every voter submits a single candidate (not a ranking) and the candidate with the most votes wins (various tie-breaking schemes can be adopted). However, plurality fails to account for a voter's relative preferences for any candidate other than its top ranked (assuming sincere voting). Other schemes such as the *Borda count* or *single transferable vote* take full rankings as input and produce a winner in a way that is sensitive to relative preferences. Among schemes that produce consensus rankings, the *Kemeny consensus* [15] is especially popular. Let $\mathbf{1}$ be the indicator function.

Definition 1. Let r, v be rankings, and $r = r_1 \cdots r_m$. The *Kendall-Tau* metric is $\tau(r, v) = \sum_{i<j} \mathbf{1}[v(r_i) > v(r_j)]$. The *Kemeny cost* of a ranking r with respect to votes $V = (v_1, \dots, v_n)$ is $\kappa(r, V) = \sum_{\ell=1}^{n} \tau(r, v_\ell)$. The *Kemeny consensus* is a ranking that minimizes the Kemeny cost.

Intuitively, $\tau(r, v)$ measures the number of pairwise disagreements (or inversions of candidate ordering) between an output ranking r and a vote v, and the Kemeny consensus minimizes the sum of such disagreements over all votes. Considerable work on computational social choice has focused on the Kemeny aggregation rule. It is NP-hard to compute [4, 10] but can be heuristically approximated in the context of web meta-search using local search and Markov chains [10]. A polytime approximation scheme (PTAS) is provided in [16], and approximation algorithms for the extension of Kemeny to partial rankings are given in [1]. Practical approaches for exact computation have also been explored [6].

Under certain assumptions the Kemeny consensus provides a maximum likelihood estimate of an underlying objective ranking [25, 20] (other such interpretations of common voting rules also exist [8, 7]). This perspective is common in the statistics and psychology literature, where various probabilistic models of ranked data have been proposed (e.g., Mallows, Luce-Plackett), along with methods for parameter estimation [11] and compact representations of distributions on permutations [9]. This view of estimating an objective ranking from noisy estimates is appropriate for certain applications, such as multi-class ensemble learning, where each classifier's ranking of labels can be combined [18, 17]. However, this motivation seems difficult to reconcile with the problem of social choice or consensus decision making for users with genuinely distinct preferences. Rank aggregation also has some interesting connections to the literature on rank learning. For example, Cohen et al. [5] focus on learning a preference function over all item pairs while given comparisons only between some of the items, and compute an item ranking that is most consistent with the learned preference function. This can be seen as aggregation where the comparisons come from different users and the learned ranking aggregates these preferences. In settings where we have little information of any single user's preferences (e.g., recommender systems), preferences of "similar" users can be aggregated to leverage sparse data to facilitate better learning. This is roughly the idea behind, say, *label ranking* [14].

3. THE UNAVAILABLE CANDIDATE MODEL

While rank aggregation methods like Kemeny can be used to determine winners, a consensus *ranking* is not needed for this purpose. While a consensus ranking can be used for other purposes, as argued above, the decision criterion and *how the ranking will be used* should motivate the aggregation method adopted. This is rarely done in the application of models from social choice to specific domains.

We now present a model that explicitly articulates one possible use of a consensus ranking. The *unavailable candidate model* has the usual goal of producing a winner from a set of candidates C, but once voter preferences are articulated, some of the candidates may become unavailable. For example, voters may be hiring committee members ranking job candidates that have just been interviewed. It is not known until an offer is extended whether a candidate will accept. Constructing a ranking of candidates allows the committee to move down the list as offers are refused to find the highest-ranked *available* candidate. In this sense, the consensus ranking acts as a *policy* that determines what to do under a wide variety of contingencies. We formalize this model and describe one natural criterion for optimization of rankings in such a model.

Let C be a set of candidates and V a preference profile.

Definition 2. A *policy* is a mapping $W : 2^C \to C \cup \{\perp\}$, where $W(S) \in S$ for all $S \neq \emptyset$ and $W(\emptyset) = \perp$. An *aggregation function under candidate uncertainty* maps preferences profiles V into policies W_V.

Policies are commonly known as *choice functions* [2]. A policy determines a winner for any set $S \subseteq C$ of available candidates and is useful when the set of *potential* candidates C is known prior to "voting" but the set of *available* candidates S is not.[1] Preference aggregation in such a setting requires the construction of just such a policy for any inputs V. To measure the quality of a policy, we assume a probability distribution P over 2^C, where $P(S)$ is the probability that S will be the set of available candidates. Drawing intuitions from Kemeny, our aim is to determine a policy that minimizes the expected number "disagreements" with voters under distribution P. Let $\text{top}(v, S)$ denote the top-ranked element of S in ranking or vote v. A policy *disagrees* with voter ℓ on S if $W(S) \neq \text{top}(v_\ell, S)$. The expected number of disagreements is given by:

$$\mathcal{D}(W, V) = \mathop{\mathbb{E}}_{S \sim P}\left[\sum_{\ell=1}^{n} \mathbf{1}[W(S) \neq \text{top}(v_\ell, S)]\right] . \quad (1)$$

We can view this as expected total dissatisfaction, where a voter is dissatisfied iff its top-ranked available candidate is not chosen.[2]

We may restrict attention to a specific class of policies $\mathcal{W}$ (see below). Given such a restriction, the optimal policy is:

$$W_V^{*\mathcal{W}} = \operatorname*{argmin}_{W \in \mathcal{W}} \mathcal{D}(W, V) . \quad (2)$$

An especially convenient class of policies are *ranking policies* (denoted $\mathcal{R}$). A ranking policy is specified by a ranking $r \in \Gamma_C$ and selects the highest ranked candidate in r from any available set S:

Definition 3. The *ranking policy* induced by r is $W_r : 2^C \to C \cup \{\perp\}$ where $W_r(S) = \text{top}(r, S)$ for any $S \neq \emptyset$.

In this work we focus on ranking policies. This is an especially natural class of policies that has an intuitive appeal. A ranking policy is specified fully by its "consensus" ranking, and is thus easily interpretable, very compact and easily communicable, and can be implemented with only trivial "online" computation. Of course, ranking policies are restrictive. Let $\mathcal{W}$ denote the class of all policies. It is not hard to see that the optimal unrestricted policy w.r.t. Eq. (2) corresponds to plurality voting for each $S \subseteq C$: $W_V^{*\mathcal{W}}(S) = \text{argmax}_{c \in S} |\{\ell : c = \text{top}(v_\ell, S)\}|$. And there exist V for which the optimal ranking policy $W_V^{*\mathcal{R}}$ is worse than the optimal unrestricted policy, i.e., $\mathcal{D}(W_V^{*\mathcal{R}}, V) > \mathcal{D}(W_V^{*\mathcal{W}}, V)$.

Example 1. Let $C = \{a, b, c\}$, $V = (abc, bca, cab)$ and suppose each candidate is unavailable with probability $p \in [0, 1)$. Any ranking in V induces an optimal ranking policy; so take $r^* = abc$. The "plurality-based" optimal policy $W_V^{*\mathcal{W}}$ has no more disagreements than W_{r^*} for any available set S; and if $S = \{a, c\}$ then r^* selects a (two disagreements) while $W_V^{*\mathcal{W}}$ selects c (one disagreement). Thus $\mathcal{D}(W_{r^*}, V) > \mathcal{D}(W_V^{*\mathcal{W}}, V)$ for any distribution with non-zero probability on $S = \{a, c\}$.

Despite this, the optimal unrestricted, plurality-based policy does not provide an explicit sense of which candidates are going to be chosen over others, nor does it admit a compact representation of the explicit mapping in non-algorithmic form from available candidates to a recommendation. A ranking policy is simple in several respects: it can be communicated to and readily implemented by human decision makers; it is also easy to understand. Finally, in some circumstances, a ranking may be needed to satisfy certain procedural, public policy, or legal requirements. For example, in the National Resident Matching Program (see [23] for a description), hospitals are required to submit a preference ranking of candidates for residency positions representing their preference over such candidates. Prospective residents similarly submit preferences rankings over hospitals, and a centralized matching algorithm (based on the Gale-Shapley approach [13]) finds a stable matching upon which hospitals base their offers. Since hospitals are institutions, their

[1] There are many other examples apart from job candidates for which this model applies: e.g., high stakes applications such as a building design or public projects, where feasibility of potential designs is expensive and "availability" is determined only after preferences are expressed; or low stakes settings such as selection of a restaurant for a group, where desired restaurants may be full.

[2] Other more nuanced notions of dissatisfaction can be incorporated into our model as discussed below. However we focus on this binary notion of disagreement for simplicity.

preferences will often be determined by a committee and generally represent the consensus preferences of a number of interested parties. The availability of candidates is not guaranteed (since residents may be matched to other hospitals), and can be assessed (presumably subjectively by the committee) in the form of the availability distribution P. Hence the need for a policy. But critically the Gale-Shapley matching algorithm requires that the hospital's submitted "policy" be in the form of a ranking.

As we see then, there are a variety of reasons to focus attention on ranking policies, and we deal exclusively with ranking policies in what follows.[3] We write r for its induced ranking policy W_r when no confusion will result. In the literature on individual and social choice, ranking policies are referred to as *rationalizable choice function* [2]. However, our motivation does not rely on the perspective that such policies are somehow intrinsically preferred to arbitrary policies as being more "sensible" or "rational." We simply view them as more convenient in certain situations; so we adopt the more neutral, descriptive term ranking policies.

The expected disagreement criterion in our model focuses squarely on "winner disagreements" and not relative preferences. This is why the plurality-based policy is the optimal unrestricted policy. Relative preferences for candidates other than a voter's top choice *do* influence the final ranking, but only in assessing the quality of the outcome when their top choice is unavailable. If a voter's top choice is available, then the disagreement objective does not depend on her ranking of other candidates. For this reason the Condorcet criterion will not be satisfied by our scheme (as we show below). However, this is not a fundamental limitation of our model. Our aim is to motivate the need to produce rankings as *output*. We can modify our model to easily account for different objectives, i.e., different notions of disagreement or quality, relative to a given available candidate set S. For example, Borda count could be used in Eq. (1) within the expectation over available sets, or some Condorcet method, or another measure of candidate quality that uses relative preferences. While the precise details of our analysis and algorithms below would change, the fundamental property of our model—namely, the motivation for rankings as output in the face of uncertain candidate availability—would remain unchanged.

We investigate a relatively simple class of distributions over available candidate sets in which each candidate is unavailable with identical, independent probability $p \in [0, 1]$.[4] Thus for $S \subseteq C$,

$$P(S) = p^{m-|S|}(1-p)^{|S|} \; . \tag{3}$$

When $p = 1$, the optimization problem is trivial (all policies are equally good). Thus we consider $p \in [0, 1)$ in the sequel. (When $p = 0$, the problem reduces to plurality voting over the entire candidate set C.) With this model, we can rewrite the expected number of disagreements, Eq. (1), given a ranking r and unavailability probability p as:

$$\begin{aligned}
\mathcal{D}_p(r, V) &= \sum_{\ell=1}^{n} \mathop{\mathbb{E}}_{S\sim P} [\mathbf{1}[\text{top}(r, S) \neq \text{top}(v_\ell, S))]] \\
&= \sum_{\ell=1}^{n} \sum_{S\subseteq C} \Pr_{S\sim P} [\text{top}(r, S) \neq \text{top}(v_\ell, S)] \\
&= \sum_{\ell=1}^{n} \sum_{S\subseteq C} p^{m-|S|}(1-p)^{|S|}\mathbf{1}[\text{top}(r, S) \neq \text{top}(v_\ell, S))],
\end{aligned} \tag{4}$$

We simplify notation, using $\mathcal{D}_p(r, v)$ to denote $\mathcal{D}_p(r, (v))$.

We now derive an explicit formula for the expected number of disagreements between any two rankings. For any $c \in C$ and rankings r, v, let $t(c, r, v)$ denote the number of candidates that are ranked above c in both r and v:

$$t(c, r, v) = |\{c' \in C \ : \ r(c') < r(c), \ v(c') < v(c)\}|. \tag{5}$$

For example, in Fig. 1, $t(d, r, v) = 2$ since only a and c lie above d in both rankings. Given output ranking r and vote v, intuitively r disagrees with v if it recommends an available candidate, say d, which by definition must be the top available candidate in r, but another available candidate lies above d in v. Elements "counted" in $t(d, r, v)$ do not contribute to the expected disagreement count (since they cannot be available if r recommends d), but the remaining elements above d in v do). This leads us to:

Lemma 1. *Let $p \in [0, 1)$, $C = \{c_1, \ldots, c_m\}$ a set of candidates, r, v any two rankings. Then*

$$\mathcal{D}_p(r, v) = \sum_{i=1}^{m} (1-p)p^{r(c_i)-1} \left(1 - p^{v(c_i)-t(c_i,r,v)-1}\right) \tag{6}$$

Proof. An illustration of the term inside the summation is provided in Fig. 1. Proofs of this and other results can be found in an extended version of the paper.[5] □

This result simplifies the objective function Eq. (4):

Corollary 2. *For any $r \in \Gamma_C$, $V = (v_1, \ldots, v_n) \in \Gamma_C^n$, and $p \in [0, 1)$, we have:*

$$\mathcal{D}_p(r, V) = \sum_{\ell=1}^{n}\sum_{i=1}^{m} (1-p)p^{r(c_i)-1}\left(1-p^{v_\ell(c_i)-t(c_i,r,v_\ell)-1}\right) \tag{7}$$

The problem of deciding for a given $d \geq 0$ whether there exists an $r \in \Gamma_C$ such that $\mathcal{D}_p(r, V) \leq d$ is clearly in NP. If $p = 0$ the problem reduces to placing the candidate with greatest number of "first place" votes at the top of r (i.e., plurality); the order of the remaining candidates does not affect the objective (see Sec. 5). We will also show below that if p is allowed to depend on m and n and get arbitrarily close to 1, then any minimizer of $\mathcal{D}_p(r, V)$ must also be a Kemeny consensus (Theorem 13), which is known to be NP-hard when $n \geq 4$ is an even integer (the complexity is not known for $n = 3, 5, 7, \ldots$). For other circumstances,

[3] An interesting question we leave unanswered: what is the worst-case loss in disagreement (or any other form of dissatisfaction) associated with the restriction to ranking policies?

[4] Generalizations to different (but still independent) probabilities p_c for different $c \in C$ is reasonably straightforward, but we use this single parameter model to keep the notation and exposition simple. We expect that correlated availability can be handled effectively for certain classes of distributions P (e.g., those expressed with graphical models of small size), but this is the subject of ongoing investigation. We make no assumptions about the source of this probability: it could be inferred from data, estimated by the aggregator, or elicited from voters (which itself raises interesting questions about elicitation protocols and incentives).

[5] See www.cs.toronto.edu/∼cebly/papers.html

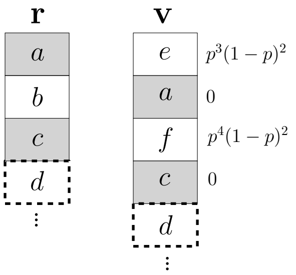

Figure 1: Determining the probability that d is recommended by r but a different candidate is preferred by v. The probability that d is the best available in r is $p^3(1-p)$, since a, b, c must be unavailable (p^3) and d available ($1-p$). Given this, the total probability of vote v disagreeing with recommendation d is determined by examining candidates *above* d: the only disagreements are with e (if available, with prob $1-p$) or f (if available and e not available, $(1-p)p$), since a, c are not available if r recommends d, and the availability of candidates below d is immaterial.

however, the NP-hardness of $\mathcal{D}_p(r, V) \leq d$ is unclear. Despite its worst case complexity in many circumstances, we explore practical approximations in the next section (both a PTAS and two empirically effective greedy algorithms). Before we address the optimization problem, we state some basic properties of our objective function.

Definition 4. Let $r = r_1 \cdots r_m$ be a ranking. The *expected number of disagreements from position i to j* is

$$\mathcal{D}_p^{(i,j)}(r, V) := \sum_{\ell=1}^{n} \sum_{u=i}^{j} (1-p)p^{u-1} \left(1 - p^{v_\ell(r_u) - t(r_u, r, v_\ell) - 1}\right).$$

$\mathcal{D}_p^{(i,j)}(r, V)$ is the expected number of disagreements when we remove candidates $C \backslash \{c_i, \ldots, c_j\}$ from consideration. To simplify notation we write $\mathcal{D}_p^i(r, V)$ for $\mathcal{D}_p^{(i,i)}(r, V)$.

Lemma 3. *For any ranking $r = r_1 \cdots r_m$ and profile V,*

$$\mathcal{D}_p(r, V) \leq n. \tag{8}$$

The expected number of disagreements from r_{k+1} to r_m is at most

$$\mathcal{D}_p^{(k+1,m)} = \sum_{\ell=1}^{n} \sum_{i=k+1}^{m} (1-p)p^{i-1}\left(1 - p^{v_\ell(r_i) - t(r_i, r, v_\ell) - 1}\right) \leq p^k n \tag{9}$$

Our objective function can also be written recursively; given $r = r_1 \cdots r_m$ we have

$$\begin{aligned}\mathcal{D}_p(r, V) &= \mathcal{D}_p^1(r, V) + \mathcal{D}_p^{(2,m)}(r \setminus \{r_1\}, V \setminus \{r_1\}) \\ &= (1-p) \sum_{\ell=1}^{n} \left(1 - p^{v_\ell(r_1) - \underbrace{t(r_1, r, v_\ell)}_{=0} - 1}\right) \\ &\quad + (1-p)p\mathcal{D}_p(r_2 \cdots r_m, V \setminus \{r_1\}),\end{aligned}$$

The base case, when there's only one candidate remaining, gives zero expected disagreements. This recursive formulation shows that if one accounts for the "contribution" r_1 makes to $\mathcal{D}_p(r, V)$, then the remaining contribution is p times the expected number of disagreements of the remaining ranking $r_2 \cdots r_m$ with respect to the votes $V \backslash \{r_1\}$. This leads to the following useful observation.

Lemma 4. *Suppose $r_1^* \ldots r_k^*$ are the top k candidates of some optimal ranking. Let r be an optimal ranking of the remaining candidates:*

$$r = \operatorname*{argmin}_{r' \in \Gamma_{C \setminus \{r_1^*, \ldots, r_k^*\}}} \mathcal{D}_p(r', V \setminus \{r_1^*, \ldots, r_k^*\}).$$

Then $r_1^* \cdots r_k^* r$ is an optimal ranking for V.

4. COMPUTING OPTIMAL RANKINGS

We now turn our attention to the problem of computing optimal rankings. We first present an integer program (IP) formulation of the ranking problem. We then describe two simple greedy approximation methods and a polynomial time approximation scheme (PTAS).

4.1 An Integer Programming Formulation

We consider the problem of computing the optimal ranking w.r.t. our objective $\min_r \mathcal{D}_p(r, V)$. We first formulate this as an IP with a quadratic objective, a polynomial number of linear constraints, and a polynomial number of binary variables. We then linearize the objective by introducing additional variables and constraints. We assume inputs C, V and $p \in [0, 1)$ are given. For notational convenience we define $[x] = \{1, 2, \ldots, x\}$ for any positive integer x.

Let $R_{ik} \in \{0, 1\}$ indicate whether candidate c_i is ranked (in the optimal ranking) at position k, and let $\mathbf{R} = \{R_{ik} : i, k \in [m]\}$. To ensure the $\mathbf{R}$ encodes a valid ranking, we require the following permutation constraints:

$$\sum_{k=1}^{m} R_{ik} = 1 \qquad \forall i \in [m] \tag{10}$$

$$\sum_{i=1}^{m} R_{ik} = 1 \qquad \forall k \in [m]. \tag{11}$$

We define variables $I_{ij} \in \{0, 1\}$ for $i, j \in [m], i \neq j$, indicating whether candidate c_i is ranked higher in the ranking than c_j, in terms of $\mathbf{R}$. We need the following constraints so that the ranking corresponding to $\mathbf{I}$ is consistent with $\mathbf{R}$:

$$1 + \sum_{\substack{j=1 \\ j \neq i}}^{m} I_{ji} = \sum_{k=1}^{m} k \cdot R_{ik} \qquad \forall i \in [m] \tag{12}$$

$$I_{ij} + I_{jq} \leq 1 + 2I_{iq} \qquad \forall i, j, q \in [m], i, j, q \text{ distinct} \tag{13}$$

$$I_{ij} + I_{ji} = 1 \qquad \forall i, j \in [m], i > j. \tag{14}$$

Constraint (12) ensures that candidate c_i's rank corresponds to $\mathbf{I}$: the summation on the lefthand side is exactly the number of candidates ranked above c_i (adding one gives us the rank of c_i), while the terms on the righthand side are all zero except for that corresponding to the rank of c_i. Constraint (13) enforces transitivity constraints on $\mathbf{I}$—namely that if c_i is above c_j and c_j is above c_q then c_i is to be above c_q. Finally constraint (14) enforces consistency on the relative ranking of c_i and c_j.

For any $t \in \{0, \ldots, m-1\}$, let $J_{it\ell} \in \{0, 1\}$ indicate whether $t(c_i, \mathbf{R}, v_\ell) = t$ (i.e., there are exactly t candidates above c_i in the ranking corresponding to $\mathbf{R}$ that also appear above c_i in v_ℓ), for each $i \in [m], \ell \in [n]$. We need constraints that encode $\mathbf{J}$ relative to $\mathbf{I}$ and V:

$$\sum_{j:v_\ell(c_j)<v_\ell(c_i)} I_{ji} = \sum_{t=0}^{v_\ell(c_i)-1} t \cdot J_{it\ell} \qquad \forall \ell \in [n], i \in [m] \quad (15)$$

$$\sum_{t=0}^{v_\ell(c_i)-1} J_{it\ell} = 1 \qquad \forall \ell \in [n], i \in [m]. \quad (16)$$

In constraint (15), the sum on the lefthand side is indexed by the set of candidates ranked higher than c_i in v_ℓ. I_{ji} will contribute to the sum iff c_j is higher ranked than c_i in $\mathbf{R}$—thus, this sum is exactly $t(c_i, r, v_\ell)$. The righthand side of (15) in conjunction with constraint (16) forces the proper $J_{it\ell}$ to take value 1. (Note that in both constraints (15) and (16), the upper limit of the sum for t should be $\min(\mathbf{R}(c_i), v_\ell(c_i))-1$, where $\mathbf{R}(c_i)$ is the rank of c_i in $\mathbf{R}$, but we cannot express $\mathbf{R}(c_i)$ without introducing additional variables.) We can now write the objective function of Eq. (7) as a quadratic objective over $\mathbf{R}$ and $\mathbf{J}$ (note that we drop the constant factor $1-p$),

$$\min_{\mathbf{R},\mathbf{I},\mathbf{J}} \sum_{\ell=1}^{n}\sum_{i=1}^{m}\left[\sum_{k=1}^{m} p^{k-1} \cdot R_{ik}\right] \left[1 - \sum_{t=0}^{\min(k,v_\ell(c_i))-1} p^{v_\ell(c_i)-t-1} J_{it\ell}\right],$$

which is equivalent to

$$\min_{\mathbf{R},\mathbf{I},\mathbf{J}} \sum_{i=1}^{m}\sum_{k=1}^{m}\left[np^{k-1} \cdot R_{ik} - \sum_{\ell=1}^{n} \sum_{t=0}^{\min(k,v_\ell(c_i))-1} p^{k+v_\ell(c_i)-t-2} \cdot J_{it\ell} \cdot R_{ik}\right].$$

The quadratic objective can be linearized by introducing a binary variable for each quadratic term. Let $Z_{ik\ell t}$ indicate whether $J_{it\ell}R_{ik} = 1$, and $\mathbf{Z}$ denote the all such variables. We impose the following constraints $\forall i, k \in [m], \ell \in [n], t \in \{0, \ldots, m-1\}$:

$$R_{ik} + J_{it\ell} \geq 2Z_{ik\ell t} \qquad \forall i, k, \ell, t \quad (17)$$

$$R_{ik} + J_{it\ell} \leq 2Z_{ik\ell t} + 1 \qquad \forall i, k, \ell, t. \quad (18)$$

The quadratic objective is linearized using $\mathbf{Z}$ to obtain the following IP:

$$\begin{aligned} \min_{\mathbf{R},\mathbf{I},\mathbf{J},\mathbf{Z}} \quad & \sum_{i=1}^{m}\sum_{k=1}^{m}\left[np^{k-1} \cdot R_{ik} - \sum_{\ell=1}^{n} \sum_{t=0}^{\min(k,v_\ell(c_i))-1} p^{k+v_\ell(c_i)-t-2} \cdot Z_{ik\ell t}\right] \\ \text{subject to} \quad & (10), (11), (12), (13), (14), (15), (16), (17), (18) \\ & \mathbf{R}, \mathbf{I}, \mathbf{J}, \mathbf{Z} \in \{0, 1\}. \end{aligned} \qquad \text{(IP1)}$$

The total number of variables is at most $nm^3 + (1+n)m^2 - mn$ and the number of constraints is $(1+2n)m^3 - m^2/2 + (2n-1/2)m$. While exact computation is desirable, this IP is quite large and will not scale to m and n of more than moderate size.[6] We turn to approximation methods in the next section to circumvent this difficulty.

4.2 Approximation Algorithms

Before presenting approximation algorithms for ranking, we first make a key observation: any candidate with at least $n/2$ top votes is a top-ranked candidate in some optimal ranking; and this holds for the restriction to any subset of candidates. We exploit this fact by having our algorithms check for such candidates and, if one exists, placing it at the top of the ranking and recursively computing the rest, thus exploiting Lemma 4.

Definition 5. Let $S \subseteq C$. $c \in S$ is a *dominant candidate* w.r.t. S, V iff $|\{\ell \in N : v_\ell(c) \leq v_\ell(c'), \forall c' \in S\}| \geq n/2$.

Note the requirements on dominant candidates are much more stringent than those on Condorcet winners.

LEMMA 5. *For any $p \in [0, 1)$, suppose there exists a dominant candidate c with respect to C and V. Then there is a ranking that minimizes $\mathcal{D}_p(r, V)$ that ranks c highest. Furthermore if c has strictly more than $n/2$ top votes then all minimizing rankings must place c at the top.*

While one might suspect that Condorcet winners can also be placed at the top of any optimal ranking, the following example shows that this is not the case.

Example 2. Let votes $V = (abcd, cbad, dbac)$. It can be seen that b is the Condorcet winner. However, when there is high probability all candidates are available (small p), b will never be chosen since it is not a top ranked candidate in any of the votes. Specifically, $\mathcal{D}_p(abcd, V) - \mathcal{D}_p(bacd, V) = 2p-1$, which is less than zero whenever $p < \frac{1}{2}$. So a lies at the top of the optimal ranking for any $p < \frac{1}{2}$.

Intuitively, the Condorcet condition fails to hold because strength of preference among available candidates is not considered in our optimization criterion (only disagreement among the top ranked available candidates is considered). But as noted above, the unavailable candidate model can be extended to account for strength of preference.

4.2.1 Greedy Algorithms

Our first greedy algorithm (see Alg. 1) is based on the objective in Corollary 2 and the observation of Lemma 5. We first check if there is a dominant candidate w.r.t. C; if so, it can be placed in the first position of an optimal ranking (Lemma 5). Otherwise, we select the top-ranked candidate r_1 by greedily choosing the candidate that induces the smallest expected number of disagreements (i.e., contributes least to the cost in Corollary 2) when placed in the top position. Once r_1 is selected, we remove it from consideration (and from the votes) and recursively choose the second-ranked candidate from $C \setminus \{r_1\}$ in the same fashion, assuming r_1 is in the first position. This is repeated for all m positions in the ranking. Note that at each step, we can easily compute $t(c, r, v)$ for any remaining candidate c and vote $v \in V$ since it depends only on the candidates ranked above c in r (hence, which have been fixed in the preceding iterations). Greedy1 has a running time of $O(nm^3)$ and has an approximation guarantee:

[6] More concise formulations and heuristics for effective computation are likely obtainable, e.g., by adapting approaches for Kemeny computation [6].

Algorithm 1 Greedy1

```
Input: $p \in [0,1)$, $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$.
  $Q \leftarrow C$
  $r \leftarrow$ empty ranking
  for $i = 1$ to $m$ do
    if $\exists c \in Q \ : \ |\{\ell \in N \ : \ v_\ell(c) \le v_\ell(c'), \forall c' \in Q\}| \ge n/2$ then
      $r_i \leftarrow c$ {$c$ is a dominant candidate}
    else
      $t(c, r, v_\ell) \leftarrow |\{r_j : j \le i-1\} \cap$
        $\{c' \in C \ : \ v_\ell(c') < v_\ell(c)\}|$, $\forall c \in Q$, $\ell \in \{1, \ldots, n\}$
      $e_c \leftarrow \sum_{\ell=1}^n 1 - p^{v_\ell(c) - t(c,r,v_\ell) - 1}$, $\forall c \in Q$
      $r_i \leftarrow \mathrm{argmin}_{c \in Q}\, e_c$
    end if
    $Q \leftarrow Q \backslash \{r_i\}$
  end for
Output: $r$
```

Algorithm 2 Greedy2

```
Input: $p \in [0,1)$, $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$.
  $e_c \leftarrow \sum_{\ell=1}^n 1 - p^{v_\ell(c) - 1}$, for all $c \in C$.
  $r \leftarrow$ sort $C$ in ascending order according to $e_c$.
Output: $r$
```

Algorithm 3 MyopicTop

```
Input: $K \ge 1$, $p \in [0,1)$, $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$
  $Q \leftarrow C$
  $i \leftarrow 1$
  $r \leftarrow$ empty ranking
  while $\exists c \in Q : |\{\ell \in N : v_\ell(c) \le v_\ell(c'), \forall c' \in Q\}| \ge n/2$ do
    $r_i \leftarrow c$ {$c$ is a dominant candidate}
    $Q \leftarrow Q \backslash \{c\}$
    $i \leftarrow i + 1$
  end while
  $(r_i, r_{i+1}, \ldots, r_{i+K-1}) \leftarrow \mathrm{argmin}_{r' \in \Gamma_Q}\, \mathcal{D}_p^{(i,i+K-1)}(rr', V)$, e.g. by solving (IP2)
  $Q \leftarrow Q \backslash \{r_i, \ldots, r_{i+K-1}\}$
  $(r_{i+K}, \ldots, r_m) \leftarrow$ arbitrarily order candidates in $Q$
Output: $r$
```

THEOREM 6. *The approximation ratio of* Greedy1 *is at most* $(1+p^2)/(1-p)^2$. *That is, if* $\min_r \mathcal{D}_p(r, V) \neq 0$,

$$\frac{\mathcal{D}_p(\mathsf{Greedy1}(p, C, V), V)}{\min_r \mathcal{D}_p(r, V)} \le \frac{1+p^2}{(1-p)^2} \ ,$$

otherwise $\min_r \mathcal{D}_p(r, V) = \mathcal{D}_p(\mathsf{Greedy1}(p, C, V), V) = 0$.

The proof of this theorem relies on a method used to prove Theorem 7 and, as discussed there, we believe this can be tightened significantly. Despite its looseness, this bound is reasonable for small p. For large p (say, $p > 0.25$), the bound is too loose to be useful.

The second greedy algorithm is even simpler: we compute the expected number of disagreements induced by each candidate assuming it is placed in the top position, then output the (ascending) sorted list of candidates as the ranking (see Algorithm 2). It can be verified that Greedy2 has running time $O(nm + m \log m)$. Unlike Greedy1, we do not recompute the disagreement score each time a candidate is placed in position.

The advantage of these greedy algorithms is that they are simple to implement and have fast running times. In our experiments, both these algorithms perform very well in terms of approximation ratio on randomly generated votes (see Section 6) with Greedy1 outperforming Greedy2.

4.2.2 A Polynomial Time Approximation Scheme

We now show that there exists a polytime algorithm that can approximate the optimal ranking in the unavailable candidate model arbitrarily well. We provide a *polynomial time approximation scheme* for computing the ranking that minimizes the expected number of disagreements with a preference profile when p is upper bounded by any constant less than 1. That is, given any $\epsilon > 0$, along with the usual inputs, the algorithm will run in polynomial time in $1/\epsilon, m, n$ and output a ranking that has an expected number of disagreements that is within a factor $1+\epsilon$ of that of the optimal solution.

The PTAS is based on the observation that the expected number of disagreements with r, $\mathcal{D}_p(r, V)$, is more sensitive to the ordering of candidates near the top of the ranking than the bottom. Indeed, candidates higher in the ranking contribute exponentially more to the expected disagreement score than lower candidates. We exploit this by finding the myopically optimal *top K candidates*, i.e., a subranking of size K that minimizes $\mathcal{D}_p^{(1,K)}(r, V)$. Since the remaining positions in r are less important, we order these $m-K$ candidates arbitrarily. The algorithm is somewhat more clever in that it also checks for dominant candidates (which must lie at the top of an optimal ranking). We show that the value of K needed to derive a PTAS depends only on constants ϵ and an upper bound on p.

The myopically optimal top K algorithm is shown in Algorithm 3. We use rr' to denote the current ranking r (which is not yet "completed"), with a ranking r' of some other candidates appended to r. The algorithm requires a method to find the top K candidates of votes V' over candidates Q. In theory this can be done in polytime by exhaustive search over all $m!/(m-K)!$ possible length K subrankings; however, a more sensible approach is to solve a modified version of the IP (IP1) in which we truncate the objective to consider only the contributions of the top K candidates to the total expected number of disagreements, obtaining:

$$\begin{aligned} \min_{\mathbf{R},\mathbf{I},\mathbf{J},\mathbf{Z}} \quad & \sum_{i=1}^{m}\sum_{k=1}^{K} \Big[np^{k-1} \cdot R_{ik} - \\ & \sum_{\ell=1}^{n} \sum_{t=0}^{\min(k,v_\ell(c_i))-1} p^{k+v_\ell(c_i)-t-2} \cdot Z_{ik\ell t} \Big] \\ \text{subject to} \quad & (10),(11),(12),(13),(14),(15),(16),(17),(18) \\ & \mathbf{R},\mathbf{I},\mathbf{J},\mathbf{Z} \in \{0,1\} \end{aligned} \qquad \text{(IP2)}$$

THEOREM 7. *Let* $d \in [0,1)$. *Let constant* $\epsilon > 0$ *be our desired accuracy. Suppose we run the algorithm* MyopicTop *with inputs* $p \in [0,d]$, C *of size* $m \ge 1$, V *of size* $n \ge 1$, *and set* $K = \left\lceil \log \frac{2}{\epsilon(1-p)^2} / \log \frac{1}{p} \right\rceil$. *Let* $r^* = \mathrm{argmin}_r\, \mathcal{D}_p(r, V) =$

2 voters	2 voters	1 voter	1 voter
a	b	c	c
b	a	a	d
c	c	b	b
d	d	d	a

Figure 2: Kemeny is not optimal for these votes.

$r_1^* \cdots r_m^*$ be the optimal ranking of C given votes V; and let $\hat{r} = \mathsf{MyopicTop}(K, p, C, V) = \hat{r}_1\hat{r}_2 \cdots \hat{r}_m$ be the ranking output by the algorithm. Then: (a) MyopicTop runs in polynomial time in m, n; and (b) if $\mathcal{D}_p(r^*, V) = 0$ then $\mathcal{D}_p(\hat{r}, V) = 0$, otherwise

$$\frac{\mathcal{D}_p(\hat{r}, V)}{\mathcal{D}_p(r^*, V)} \leq 1 + \epsilon.$$

Thus algorithm MyopicTop is a PTAS for the ranking problem. The proof demonstrates an $O\left(m^{\log_{\frac{1}{d}} \frac{2}{\epsilon(1-d)^2}}\right)$ running time for the algorithm. We bound p by an (arbitrary) upper bound d to remove dependence of p on m and n, so that K does not depend on m, n; such a dependence would render the algorithm no longer polytime (note that as p increases, K becomes larger). The exponential dependence on $\log(1/\epsilon)$ is sufficient for a PTAS since ϵ is a fixed (but arbitrary) constant.

5. RELATION TO OTHER SCHEMES

We now consider relationships between the consensus ranking induced by the unavailable candidate model and two popular voting rules, the Kemeny consensus [15] and plurality. Recall the Kemeney consensus from Defn. 1. For any ranking $r = r_1 \cdots r_m$ and preference profile V, the Kemeny cost can also be written recursively:

$$\kappa(r, V) = \sum_{\ell=1}^{n} \sum_{i=2}^{m} \mathbf{1}[v_\ell(r_1) > v_\ell(r_i)] + \kappa(r_2 \cdots r_m, V \setminus \{r_1\}).$$

The base case is when only one candidate remains and of course the Kemeny cost is zero. Thus if we are given the top candidate of a Kemeny consensus, computing the best remaining ranking gives us a full Kemeny consensus. This leads to the following observation (similar to Lemma 4):

Lemma 8. *Let $r_1^* r_2^* \cdots r_k^*$ be the top k candidates of some Kemeny consensus with respect to votes V. Then computing the Kemeny consensus over the remaining candidates,*

$$r = \operatorname*{argmin}_{r' \in \Gamma_{C \setminus \{r_1^*, \ldots, r_k^*\}}} \kappa(r', V \setminus \{r_1^*, \ldots, r_k^*\}),$$

gives us a Kemeny consensus $r_1^* \cdots r_k^* r$.

Given our objective—which aims to maximize the number of voters who consider the selected candidate best among available candidates—not surprisingly our scheme corresponds to plurality voting when $p = 0$:

Theorem 9. *Let $p = 0$, and $r^* = \operatorname{argmin}_r \mathcal{D}_p(r, V)$. Then the top candidate in r^* is a candidate with the maximum number of first-place votes (i.e., a winner under plurality).*

Obviously, when $p = 0$, an arbitrary ordering of candidates will suffice for positions 2 and higher in the optimal ranking. As discussed above, different objectives that, say, account for relative preferences, can be incorporated into the unavailable candidate model.

There are tight connections between the optimal ranking in our model and the Kemeny ranking as well. First, we note that the models are not the same:

Lemma 10. *For any $p \in (0, 1)$, there exist candidates C and votes V such that the Kemeny consensus K^* does* not *minimize $\mathcal{D}_p(\cdot, V)$.*

A counterexample is shown in Fig. 2, where the two Kemeny consensus rankings for the votes shown are *abcd* and *bacd*; but only $r^* = abcd$ minimizes $\mathcal{D}_p(\cdot, V)$ for any $p \in (0, 1)$. However, for p close to 1 (where closeness is a function of m and n), we show that any optimal ranking in our model must be Kemeny consensus. This implies that at least one Kemeny consensus is optimal; but the full converse is not true as we have shown. Not every Kemeny consensus needs to be optimal under our model. However, we show that any Kemeny consensus is a good approximation of the optimal ranking. Intuitively, as p gets closer to 1, candidates lower in the ranking contribute more to the expected number of disagreements. Thus, the optimal ranking must get the *entire* ranking "right." Consequently, the unavailable candidate model can be used justify Kemeny in certain scenarios. This can be seen as an alternative justification—at least in some circumstances—of the use of the Kemeney consensus as an aggregation technique for "subjective" rankings, just as Young's maximum likelihood model [25] or Mallow's model [20] justifies its use for estimating an "objective" ranking under suitable assumptions.

Theorem 11. *Fix $m, n \geq 1$. For any set of m candidates C, n votes V, positive $\epsilon < \frac{2}{nm(m-1)+2}$ and $p > (1-\epsilon)^{\frac{1}{m-1}}$, we have that:*

1. *Any ranking that minimizes the expected number of disagreements is also a Kemeny consensus.*
2. *Any Kemeny consensus K^* has the property*

$$\frac{\mathcal{D}_p(K^*, V)}{\min_r \mathcal{D}_p(r, V)} \leq \frac{1}{1-\epsilon}\ .$$

Hence for values of p very close to 1, finding a Kemeny consensus can be reduced to minimizing the expected number of disagreements $\mathcal{D}_p(\cdot, V)$ in the unavailable candidate model. This demonstrates the NP-hardness of our ranking model for large values of p (see Corollary 12). But while the converse fails to hold, any Kemeny consensus provides a very good approximation of our objective, which gets better as p gets closer to 1. In fact, this theoretical phenomenon is clearly demonstrated in our empirical results (next section).

Corollary 12. *Let p be large as in Theorem. 11. For a given $x \geq 0$, deciding whether there exists a ranking r such that $\mathcal{D}_p(r, V) \leq x$ is NP-complete for even integers $n \geq 4$.*

Theorem 11 shows that the Kemeny consensus approximately minimizes the expected number of disagreements when p is close to 1. But for small values of p this is generally not the case.

Theorem 13. *For any $p \in [0, 1)$, let K^* be a Kemeny consensus and $r^* = \operatorname{argmin}_r \mathcal{D}_p(r, V)$ any optimal ranking. If $\mathcal{D}_p(r^*, V) \neq 0$, then:*

$$\frac{\mathcal{D}_p(K^*, V)}{\mathcal{D}_p(r^*, V)} \leq \frac{2}{(1-p)^2}\ ; \qquad (19)$$

k voters	k voters	$2k-1$ voters
c	d	a
b	b	b
a	a	c
d	c	d

Figure 3: Poor performance of Kemeny as $p \to 0$.

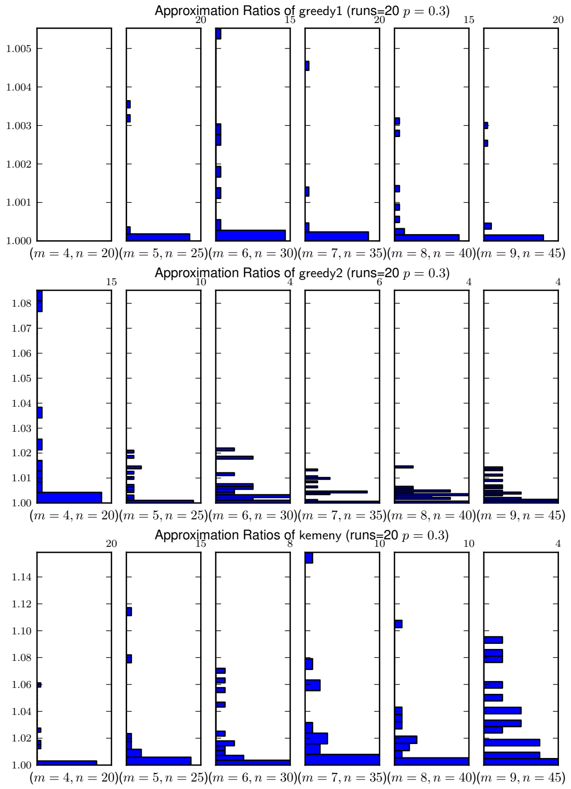

Figure 4: Histogram of approximation ratios (20 instances) for small m, n, $p = 0.3$. An empty plot means all output rankings are optimal.

otherwise $\mathcal{D}_p(r^*, V) = \mathcal{D}_p(K^*, V) = 0$.

This bound is loose for larger values of p. However, for small p, we can show that the Kemeny consensus can be close to a factor of two worse than the optimal ranking.

Example 3. Consider the votes shown in Fig. 3. It can be seen that the Kemeny consensus K^* is *bacd* (note that b is the Condorcet winner). For $k \geq 2$ and $p < 3/4$ the optimal ranking in our model is $r^* = abcd$. Thus,

$$\frac{\mathcal{D}_p(K^*, V)}{\mathcal{D}_p(r^*, V)} = \frac{4k - 1 + 2kp + kp^2}{2k + 4kp + kp^2} = \frac{2 + p + p^2 - 1/(2k)}{1 + 2p + p^2/2}$$

which approaches 2 from below as $p \to 0$ and $k \to \infty$.

6. ALGORITHMIC EXPERIMENTS

We perform experiments that examine the performance of our greedy algorithms and the Kemeny consensus. We compare the outputs r of these ranking schemes on random problems by examining the ratio of expected number of disagreements relative to the (minimum) number obtained by the optimal ranking r^*: $\mathcal{D}_p(r, V)/\mathcal{D}_p(r^*, V)$. We also examine the computational performance of the greedy algorithms.

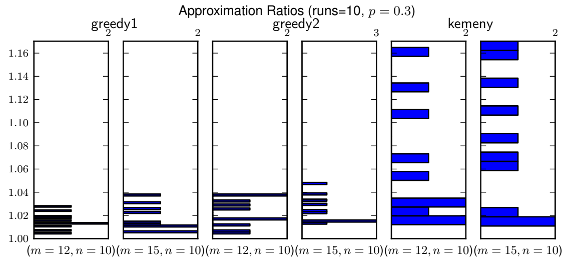

Figure 5: Histogram of *upper bounds* on approximation ratios (10 instances) for large m, $n = 10$, $p = 0.3$.

We generate random problem instances as follows given fixed numbers m of candidates and n of voters. We first randomly partition voters into "types," with voters of the same type having their votes generated by a noisy realization of a common ranking. There are $\log_2 n$ types, and each type θ is characterized by a "score vector" $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ (each score drawn from an exponential distribution with rate $\lambda = 3$). A voter of type θ has her vote generated as in the Luce-Plackett model [19, 21]: candidate c_i is placed in position 1 with probability proportional to θ_i; the selected candidate is eliminated, and the remaining candidates are selected for position 2 in the same way; and this is repeated until all m candidates are placed.

In our first series of experiments, we fix the value of candidate unavailability at $p = 0.3$ and vary m and n, generating random instances for each m, n-pair. Fig. 4 shows the histogram of the approximation ratios of both greedy algorithms and Kemeny relative to the optimal ranking for small values of m (in the range 4–9) and large values of n (ranging from 20 to 45). Fig. 5 shows the same for large values of m ($m = 12, 15$) and $n = 10$. and Fig. 6 shows the average ratios for both small and large m values. For small m, 20 instances were generated and the optimal ranking was computed by exhaustive search. For large m, 10 instances were generated and CPLEX 11.1.1 was used to solve our IP (IP1) with a time limit of 5 minutes for $m = 12$ and 10 minutes for $m = 15$, and its lower bound on expected disagreements used. Thus the approximation ratios for large m are in fact upper bounds, with the true ratio potentially better than the ratio relative to the lower bound that we compute. Kemeny rankings were computed using the IP formulation of [6]. Computationally, the greedy algorithms solve all problems in well under one second, while the Kemeny IP is bit slower but still under a second for problems of this size (though Kemeny optimization takes well over a minute for $m \geq 100$, while the greedy methods remain under one second).

These results indicate that Greedy1 offers an extremely good approximation algorithm and often finds the optimal solution, and does so very quickly. This also suggests the approximation ratio bound in Theorem 6 is overly pessimistic. Relatively speaking, Greedy2 performs worse, though it still provides very good approximations with an asymptotically better running time. The Kemeny consensus performs worst w.r.t. expected disagreements in our model (often quite poorly, note the differences in y-axis scale), but does also provide a reasonable approximation in some cases. Not surprisingly its performance gets worse as the number of voters increases.

	Parameters (m, n) varies			
$p = 0.3$	$(4, 20)$	$(5, 25)$	$(6, 30)$	$(7, 35)$
greedy1	**1.00000**	**1.00036**	**1.00081**	**1.00033**
greedy2	1.01190	1.00597	1.00621	1.00450
kemeny	1.00554	1.01546	1.01665	1.02429
	$(8, 40)$	$(9, 45)$	$(12, 10)$	$(15, 10)$
	1.00046	**1.00034**	**1.01587**	**1.01816**
	1.00355	1.00424	1.02276	1.02572
	1.01605	1.03074	1.06411	1.08512

Figure 6: Average approximation ratio of different algorithms for $p = 0.3$ (20 instances for $m \leq 9$, 10 instances for $m > 9$, average *upper bound* is shown for $m = 12, 15$.)

	Parameters $m = 8, n = 40$, p varies			
	0.01	0.05	0.10	0.20
greedy1	**1.00003**	**1.00000**	**1.00004**	**1.00030**
greedy2	1.00018	1.00032	1.00101	1.00251
kemeny	1.01178	1.04776	1.02528	1.03288
	0.5	0.70	0.90	0.95
	1.00171	**1.00385**	1.00667	1.00698
	1.00615	1.00630	1.01087	1.00817
	1.01435	1.00386	**1.00039**	**1.00025**

Figure 7: Average approximation ratio of different algorithms for $m = 8, n = 40$ (20 instances for each p). The results corroborate Theorem 11.

In a second set of experiments, we vary values of p while fixing $m = 8, n = 40$. Similar histograms are shown in Fig. 7 and average approximation ratios are plotted in Fig. 8. Some interesting patterns emerge in the histograms: the greedy algorithms perform extremely well for p up to 0.7 (with Greedy1 dominating), but get relatively worse as p approaches 1. This is due to the fact that the greedy algorithms focus on placing the "best candidates" in the top positions at the potential expense more disagreements near the bottom of the ranking. As p gets larger, disagreements near the bottom contribute more to expected cost. Still the greedy algorithms provide very good approximations, particularly Greedy1. The Kemeny consensus, by contrast, improves as p approaches 1, and for values of $p \geq 0.9$ dominates the greedy algorithms. This corroborates Theorem 11, which states that, as p approaches 1, Kemeny will offer a very good approximation of the optimal ranking and additionally, for p *very* close to 1, the optimal ranking is a Kemeny consensus.

7. CONCLUSION AND FUTURE WORK

We have introduced the unavailable candidate model, a novel model of social choice that provides one possible rationale for computing a consensus ranking of voter preferences rather than a single decision. Our model gives rise to a principled objective function for rank aggregation, one that differs from classical rank aggregation rules, but bears some strong connection to such methods, including the plurality and Kemeny voting rules. We have provided exact algorithms as well as approximation algorithms that exploit the property that items higher in the ranking are more "important" with respect to their contribution to the objective function. Apart from theoretical bounds on performance, our experiments indicate that the greedy algorithms provide excellent approximations to the optimal ranking—especially Greedy1, which frequently finds the optimal ranking—and are computationally effective. Empirical evidence also corroborates the theoretical connections we draw between our model and the Kemeny consensus.

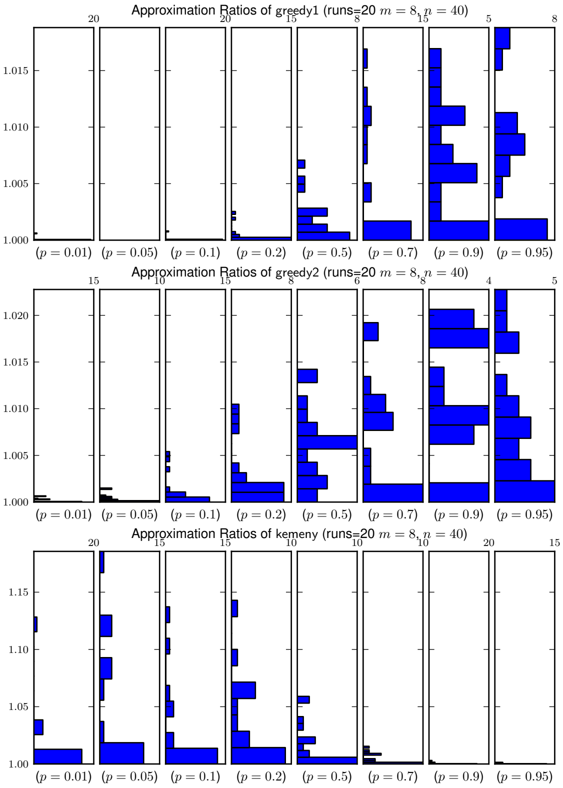

Figure 8: Histogram of approximation ratios (20 instances) for various p, and fixing $m = 8, n = 40$. An empty plot means all output rankings are optimal.

The unavailable candidate model is an illustration of the use of a specific decision criterion to derive a specific preference aggregation model, and one that naturally gives rise to consensus rankings rather than single winners. As emphasized above, there are a number of other decision models can justify the use of consensus rankings. The broader lesson is that the specific decision model should be used to determine the choice, or derivation, of social choice or rank aggregation methods that maximize the desired decision criterion.

Future work includes looking at various extensions of the unavailable candidate model. One of the most important is to incorporate criteria other than "binary disagreement" with the chosen candidate from among available candidates, using voter relative preference over all available candidates to derive a ranking. Other interesting and realistic distributions over available candidates are being investigated, as well as methods for 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 proving tighter approximation bounds for the greedy algorithm remains open.

8. REFERENCES

[1] N. Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *18th Annual Symposium on Discrete Algorithms (SODA)*, pp.415–424, New Orleans, 2007.

[2] F. Aleskerov. Categories of Arrovian voting schemes. In *Handbook of Social Choice and Welfare, Vol. 1*, K. J. Arrow, A. K. Sen, K. Suzumura (eds), Elsevier, Amsterdam, 2002.

[3] K. J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58:328–346, 1950.

[4] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:57–165, 1989.

[5] W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1998.

[6] V. Conitzer, A. J. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In *21st National Conf. on Artificial Intelligence (AAAI)*, pp.620–626, Boston, 2006.

[7] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *21st International Joint Conf. on Artificial Intelligence (IJCAI)*, pp.109–115, Pasadena, CA, 2009.

[8] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *21st Conf. on Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, pp.145–152, 2005.

[9] P. Diaconis. *Group Representations in Probability and Statistics*. Inst. Math. Statistics, Hayward, CA, 1988.

[10] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *10th International World Wide Web Conf. (WWW)*, pp.613–622, Hong Kong, 2001.

[11] M. A. Fligner and J. S. Verducci (eds.). *Probability Models and Statistical Analysis for Ranking Data.* Springer, NY, 1993.

[12] W. Gaertner. *A Primer in Social Choice Theory.* Oxford, 2006.

[13] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69:9–14, 1962.

[14] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16-17):1897–1916, 2008.

[15] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[16] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *39th Symposium on Theory of Computing (STOC)*, pp.95–103, San Diego, 2007.

[17] A. Klementiev, D. Roth, K. Small, and I. Titov. Unsupervised prediction aggregation. In *NIPS 2009 Workshop: Learning with Orderings*, Whistler, BC, 2009.

[18] G. Lebanon and J. D. Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *19th International Conf. on Machine Learning*, pp.363–370, Sydney, 2002.

[19] R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis.* Wiley, 1959.

[20] C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[21] R. Plackett. The analysis of permutations. *Applied Statistics*, 24:193–202, 1975.

[22] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *13th International Conf. on Knowledge Discovery and Data Mining (KDD)*, pp.570–579, San Jose, CA, 2007.

[23] A. E. Roth. The National Residency Matching Program as a Labor Market *Journal of the American Medical Association*, 275(13):1054-1056

[24] A. K. Sen. *Collective Choice and Social Welfare.* Holden-Day, San Francisco, 1970.

[25] P. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9:51–64, 1995.

Acknowledgements

Thanks to the anonymous reviewers for helpful comments. This research was supported by the Natural Sciences and Engineering Research Council of Canada.