Group Decision Making with Partial Preferences

by

Tyler Lu

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Group Decision Making with Partial Preferences

Tyler Lu
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2015

Group decision making is of fundamental importance in all aspects of a modern society. Many commonly studied decision procedures require that agents provide full preference information. This requirement imposes significant cognitive and time burdens on agents, increases communication overhead, and infringes agent privacy. As a result, specifying full preferences is one of the contributing factors for the limited real-world adoption of some commonly studied voting rules.

In this dissertation, we introduce a framework consisting of new concepts, algorithms, and theoretical results to provide a sound foundation on which we can address these problems by being able to make group decisions with only partial preference information.

In particular, we focus on single and multi-winner voting. We introduce *minimax regret (MMR)*, a group decision-criterion for partial preferences, which quantifies the loss in social welfare of chosen alternative(s) compared to the unknown, but true winning alternative(s). We develop polynomial-time algorithms for the computation of MMR for a number of common of voting rules, and prove intractability results for other rules. We address *preference elicitation*, the second part of our framework, which concerns the extraction of only the relevant agent preferences that reduce MMR. We develop a few elicitation strategies, based on common ideas, for different voting rules and query types.

While MMR can be applied in a distribution-free context, in many practical environments decision makers have access to historical datasets of, and probabilistic knowledge of agent preferences. To leverage such information, we first address the problem of learning probabilistic models of preferences from pairwise comparisons—the building block

of many preference structures—for which previous techniques cannot handle. Then we extend our framework to a *multi-round* elicitation process that leverages probabilistic models to guide and analyze elicitation strategies.

We empirically validate our framework and algorithms on real datasets. Experiments show our elicitation algorithms query only a fraction of full preferences to obtain alternative(s) with small MMR. Experiments also show our learning algorithms can learn accurate mixture models of preference types, which we then use to guide the design of one-round top-k elicitation protocols.

Dedication

I dedicate this thesis to my family.

Acknowledgements

First and foremost, I would like to thank my advisor Craig Boutilier for assiduously guiding me during my PhD. Craig is a leading researcher in the AI community who is always enthusiastic about informing me of his new ideas, helping me push the envelope on my research, advising me throughout the research process, and engaging me in the broader research community.

I would also like to thank members of my PhD committee: Alan Borodin, Martin Osborne and Rich Zemel. They were always available to talk to and provided advice and guidance throughout my PhD. They also sat in on my practice talks, listened carefully, took notes and had excellent questions, astute observations, and some great all around advice.

I had the pleasure to work with, and to co-author papers with some of the best minds in all of computer science. They include Ariel Procaccia, Ioannis Caragiannis, Moshe Tenneholtz, Or Sheffet, Pingzhong Tang, Reshef Meir and Simi Haber.

I am grateful to my colleagues in the research community who have provided plenty of feedback, ideas and support during my PhD. They include my co-authors, Jerome Lang, Lihi Dery, Lirong Xia, Meir Kalech, Tuomas Sandholm, Vincent Conitzer and many others whom I've had plenty of productive and enjoyable conversations with.

I would also like to thank the AI faculty, including Geoff Hinton, Rich Zemel and Sheila McIlraith, who have taken an interest in my career, my research, and who have encouraged and cheered me on. It is this kind of collegial and caring environment that has boosted my self-confidence, ambitions and my drive to think big and do great things. I want to thank my fellow students who I can, and *do*, talk to about anything. There is no doubt that the people and culture that exists here is the reason the department is ranked so highly and one that has cultivated some of the leading researchers throughout the years. In fact, I would have to admit that sometimes my practice talks attracted a more intense and accomplished audience than the one in my actual talk at the actual conference. Luckily, they were all on my side.

Being part of the Department of Computer Science during the past five years has been a great blessing. There have been some great research produced here, some great people flowing through here, and some great projects that have impacted society at large, sometimes through starting companies. There has never been a more exciting time to be here.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

People, societies, and organizations routinely make decisions that impact not just a single entity, but multiple individuals and stakeholders. Whether this involves scheduling a meeting or nations electing their leaders, decisions must be made in the presence of diverse preferences amongst the agents. In practice, such decision processes typically involve informal deliberations, limited forms of voting, or outright dictatorship. But with the recent rise in ubiquitous forms of electronic communications such as social media and smartphones, there are greater opportunities to apply novel approaches and algorithms to these group decision problems so as to enable more efficient and desirable outcomes. This creates the need to develop more intelligent, transparent, and sophisticated mechanisms capable of reducing the time and effort required by users to specify their preferences, as well as increasing the overall satisfaction with selected decisions.

One of the long-standing priorities in artificial intelligence is the development of systems that can autonomously make or otherwise assess decisions for one or more underlying agents. Making decisions for these *multi-agent systems* is the primary focus of this thesis. The field of decision making for groups of individuals, typically studied in political science and economics, is termed *social choice*. One of the main interests within social choice is the study of voting rules that take the preferences of individuals as input, such as rankings over all given *alternatives*, and output a recommended choice that reflects some notion of consensus or compromise. In fact, one usually thinks of social choice, from the perspective of social scientists, as being primarily concerned with theoretical aspects of elections and political matters. However, with the success of recommender systems and the ease with which we can state our preferences (such as rating products or comparing options), it is becoming more realistic to conduct both high-stakes and low-stakes social choice events electronically. In order to realize this shift, progress must be made in both the theory and practice of *computational social choice*.

The major bottleneck in both single or multi-agent decision making is the availability of preference information. We seek to alleviate this burden on the agents, by providing a protocol under which partial preference information is *elicited* incrementally, and decisions or social choices can be made or recommended even with such partial preferences. Such decisions should be robust—in the sense that worst-case guarantees can be made about their quality. As discussed below, the quality measure is typically implicit in the underlying social choice mechanism, as it usually measures the social welfare of its selected alternative. Preference information should be elicited cleverly—preferences and agents that are or become irrelevant to the best decision should be pruned from queries. Furthermore, elicitation should be adaptive—using the history of questions and answers to update the best queries.

By reducing the number of questions and amount of information elicited, the decision maker may be able to arrive at a "good enough" decision faster and with less cognitive and communication effort from agents while better protecting their privacy. We also require that algorithms for these problems be practical in the sense that they are relatively simple to implement, have good time and space complexity, and have good performance on real world datasets: in terms of computation, decision quality and the amount of information elicited. The primary aim of this thesis is to address these issues. In particular, we seek to develop models and algorithms to reduce the cognitive and communication burdens on agents or voters engaged in social choice decisions.

Existing research has not provided a principled approach under which decisions can be made, recommended or assessed with arbitrary collections of partial preference information. Past work has introduced and studied notions of *necessary* and *possible* winners, and while these ideas are suitable when winners can be fully determined from given partial preferences, they have no relevance in the vast majority of practical scenarios when such partial preference information cannot fully determine the winners. Simple elicitation strategies involving repeatedly querying an agent's next favourite alternative have also been proposed and studied. Such approaches, however, are not fully incremental (querying all agents at once), rely on the problematic concepts of necessary and possible winners, and are restrictive in the types of queries considered.

We focus on making group decisions that do not involve the transfer of money, or more generally the transfer of utility. We focus on settings where agents have underlying preference rankings over a finite set of alternatives, and we assume that partial preference information consists of a consistent set of pairwise alternative comparisons. For example, this may model a situation where each member of a hiring committee (the agents) has an ordinal ranking over the set of candidates (the alternatives). This ranking usually

reflects a personal preference or an objective evaluation of the candidates. In the hiring committee example, each member's ranking can reflect both: objective evaluations of candidates' level of job skills and experience, and personal preferences regarding candidates' cultural fit. Then a social choice mechanism, which receives as input the preference rankings of each committee member, outputs a decision. In the example, this may be a single candidate, or even a selection or slate of candidates to hire.

A social choice mechanism typically reflects some notion of consensus. For example a mechanism may score each candidate in a ranking by the position of the ranked candidate, and then sum the scores across all agents' rankings to obtain a total score for each candidate. Then the candidate with the highest total score is selected. This was effectively the technique, now known as the *Borda count*, used by the French Academy of Sciences in 1770 after being introduced by de Borda [39]. More generally, the Borda count belongs to a class of voting schemes known as *positional scoring rules* which assigns arbitrary non-increasing scores to each of the rank positions. Such rules are based on an implicit assumption that each rank position corresponds to a *utility* value the agent has when the candidate ranked at the said position is chosen by the voting scheme. And the candidate with the maximum total positional score is selected. This type of voting scheme, roughly speaking, implements notions of utilitarianism in the theory of politics. There are also social choice schemes (e.g., maxmin fairness), that implement notions of egalitarianism, which is the idea that the utility of the least well off agent (in this case the agent whose rank position of the chosen alternative is the largest) should be maximized.

Many of the social choice mechanisms have an underlying objective that is maximized or minimized (such as any positional scoring rule) reflecting some social desiderata. Still other rules are more algorithmic and procedural based, such as Single Transferable Vote or Dodgson's rule. Such rules, however, can be motivated from axiomatic foundations, which in itself has a long history in social choice—starting with Arrow's Impossibility Theorem [4]. Axiomatic approaches serve another avenue from which to devise social choice rules with desired properties. These and other rudimentary aspects of social choice can be found in an introductory text (see e.g., [51]).

In this dissertation, we generally do not assume preferences are restricted. Agents may express all $m!$ possible rankings (where m is the number of alternatives) independently of the rankings of the other agents. In some situations restricted preferences may arise—a common example are single-peaked preferences. For example, in elections where candidates can be ordered from left to right on the political spectrum, then it is likely that the preferences of a particular agent is relatively stronger for a candidate near the agent's ideal leaning on the political spectrum.

1.1 Overall Contributions

The purpose of this PhD thesis is to advance the theory and practical design of decision support systems for social choice. In particular, we develop models and algorithms for making and assessing group decisions with only partial preference information, and algorithms for the effective elicitation of only the most relevant preference information. To make progress towards these two goals, we address three sub-goals:

1. Reducing the cognitive and communication burden on agents involved in a group decision;

2. Developing learning algorithms to learn statistical models of rankings with arbitrary pairwise preference data; and

3. Integrating statistical models of rankings into group decision support so as to further improve decision quality while reducing user cognitive and communication burden.

We now detail below these three points.

Reducing Interaction Burdens on Users

In this research thread, we are interested in making group decision mechanisms more practical for agents. One of the main reasons that commonly studied social choice schemes—which typically require participants to provide a full ranking over all alternatives—are not deployed in practice is the cognitive and communication burden that they impose on users. Individuals or organizational entities would like to avoid the exercise of thinking about and communicating a full ranking if it is not necessary. Moreover, agents may be hesitant to reveal their preferences for reasons of privacy or confidentiality. This is in part why *Plurality voting*, where users simply vote for their favourite candidate, is by far the most popular voting rule used in practice.

Past work has done little to address this issue [76, 78, 123, 32, 121], and one of the aims of this thesis work is to relieve voters of this burden while maintaining the benefits of rank-based voting schemes. Allowing users to flexibly specify partial preferences—which may come in the form of a ranking of their top few choices, or simple pairwise comparisons of a few options—would help reduce cognitive burden. Given this partial preference information, there is the problem that the decision maker still needs to assess and make good group decisions. Furthermore, not all partial preferences are equally useful—being able to selectively query or elicit the most relevant preference information

can reduce the time and effort required of users, while still allowing one to reach a high quality or optimal decision. In order to achieve these goals, new conceptual frameworks must be developed to formalize notions of voting with partial preferences and elicitation of the relevant preferences. Moreover these notions must be tied to a solid decision-theoretic foundation for the underlying decision task.

Two of the main concepts developed in this thesis are those of *robust social choice* and *vote elicitation*. Robust social choice refers to making a group decision under partial preferences that is guaranteed, regardless of what the uncertainties may be in the worst-case, to be as close as possible to the true optimal choice. It provides a worst-case guarantee in the sense that regardless of how the partial preferences are "completed," the quality level of the recommended decision is above some tolerance. *Vote elicitation* involves intelligently asking a small number of preference queries, such as "agent 3, do you prefer A to B?" or "agent 4, what is your second favourite item?" such that a reasonably good choice outcome can be identified using robust social choice methodologies. Such elicitation strategies can operate incrementally, finding both the right agent and preferences to elicit. The ultimate goal is to elicit only the relevant preferences that lead to a good and robust group decision.

In Chapter 3, we introduce a formal, quantifiable concept of robust social choice based on the notion of *minimax regret*. Minimax regret has previously been successfully applied in the context of single-agent decision support [12, 109, 122, 13, 14, 18, 15, 19]. Roughly speaking, minimax regret is a decision criterion that aims to minimize the worst-case utility/satisfaction loss that can possibly be incurred due to the uncertainty of incomplete preferences. This measure of utility is usually implicit in common social choice rules. The alternative that corresponds to the minimax regret criterion is guaranteed to be within some ε of the actual voting outcome given the underlying full, but unobserved, preferences. This minimax criterion has a number of advantages over previous approaches of dealing with partial preferences, as we discuss in later chapters. We develop polynomial time algorithms—with respect to several popular voting rules—for computing the maximum regret of any alternative, and consequently, for identifying the minimax regret optimal alternative. We then show how minimax regret can be used to effectively drive incremental preference elicitation and devise several heuristics for this process. Despite worst-case theoretical results showing that many common voting rules require nearly complete voter preferences to determine winners, we demonstrate the practical effectiveness of regret-based elicitation for determining both approximate and exact winners on several real-world datasets.

In Chapter 4, we extend the applicability of our framework to a recently popular

multi-winner voting scheme known as proportional representation [25, 92]. This form of *slate optimization*, where we select a subset of at most K alternatives that maximizes a notion of total satisfaction, also fits within the robust social choice and incremental elicitation model. The social choice objective, however, is NP-hard to optimize even with full preference information [83]. Thus in the case of partial preferences, we develop a greedy algorithm that finds an approximately minimax regret optimal slate (subset of candidates). We also exploit *conditional max regret*, a type of score used to approximate the true max regret, as a stopping criterion, and just as importantly, we use it to drive the elicitation strategy. Experimental results show the effectiveness of both the greedy algorithm and the elicitation heuristic.

Learning probabilistic models of rankings

Historical preference data obtained from users, which are now plentiful due to user tracking technologies, are motivating researchers to develop algorithms that learn statistical models of user preferences. Such methods can be used for a variety of purposes, ranging from recommender systems to consumer product design to social choice. In social choice, probabilistic models of populational preference rankings can be used to improve both decision recommendation/assessment and elicitation strategies. However, learning models of rankings or permutations represents a challenge as there are $m!$ possible rankings, where m is the number of alternatives. Many specialized unimodal models of rankings exist, including Plackett-Luce, Thurstone and Mallows (see Marden [88] for an overview). Such models typically require much fewer parameters than the $m!-1$ probabilities needed to specify a full ranking distribution. Having a mixture of such unimodal distributions allows us to model preference diversity around clusters of rankings, each of which is defined by a modal or centre ranking. Such a mixture distribution still is quite manageable in the number of parameters required.

While past work has focused on developing algorithms for learning such mixtures [24, 95, 90], it has only been developed for very restrictive classes of preference data. Such preference structures include, for example, top- or bottom-k preferences (where agents specify a ranking of their top or bottom k most or least preferred alternatives). But such algorithms do not apply to a very fundamental and ubiquitous class of preferences: arbitrary sets of pairwise comparisons. Pairwise comparisons are fundamental building blocks of higher order preference structures. For example, a top-k or bottom-k partial preference can be reduced to a set of pairwise comparisons. The converse is not necessarily true: a single pairwise comparison (assuming three or more alternatives) cannot be represented with either a top- or bottom-k preference structure. Hence, it

is critically important to develop algorithms that can learn mixture models given such training data. Furthermore such learning algorithms should be practical and capable of scaling into the hundreds of alternatives. Algorithms that have been proposed, however, cannot learn with more than a few dozen alternatives [24, 90, 95, 79, 62].

Such is the aim of Chapter 5, where we present learning algorithms for training data consisting of arbitrary sets of pairwise comparisons. In particular, we develop the first algorithms for learning mixture of Mallows models. At the heart of our technique is a new algorithm, the *generalized repeated insertion model (GRIM)*, which allows sampling from arbitrary ranking distributions, and conditional Mallows models [87] in particular. We also show that exact sampling from a conditional Mallows is hard in general, but that the approximate samplers we develop are exact for many of the previously studied restrictive preference structures. We use these sampling methods to derive algorithms for evaluating log-likelihood, learning Mallows mixtures and for non-parametric estimation. We also show the effectiveness of our algorithms on several real-world datasets. While the focus of our learning algorithms is on a specific, and popular, form of ranking distribution known as the Mallows model, many of the techniques can be readily extended to other unimodal component distributions such as Plackett-Luce.

Leveraging preference learning

Equipped with learning algorithms for inferring populational preference models, we can naturally leverage such models to support social choice when we only have partial preferences. Two of the main applications to social choice settings are: developing better elicitation strategies (hence, further reducing user thinking and communication and better protecting privacy) by exploiting probabilistic preference predictions using learned models; and computing group decisions that aim for good *expected* quality, in addition to, or as an alternative to, robust minimax optimal decisions.

While incremental elicitation schemes will generally minimize the amount of preference information required to make a "good enough" decision, these tend to require repeated rounds of interaction from participants. For example, consider the hiring committee example, and suppose that the committee members are distributed across different geographical regions and must report their preferences electronically. An incremental elicitation procedure would, say, query member A (for a pairwise comparison), then wait for A's response. Then it must query member C, and again wait for C's response. This process of sequentially querying pairwise preferences from agent to agent introduces costly interruption overhead when each committee member must wait for an individual's response before continuing. On the other hand, when querying for full rankings, the de-

cision maker can simply ask each member, in parallel what their preference rankings are (as the queries are not dependent on previous responses, these may be asked in parallel).

One can think of these elicitation approaches as spanning multiple *rounds* where in each round a batch of queries is presented in parallel to a subset of agents. Once all agents in a round have responded, the elicitation strategy will generate the next set of queries for the next round. Hence, the queries in each round are only dependent on the responses from all prior rounds and not dependent on the responses within that round.

There is also a trade-off to be made between the number of rounds and the amount of information elicited. Fewer rounds of elicitation leads to a reduction in the overall interruption cost but it results in requiring relatively more preference information. On the other hand, increasing the number of rounds leads to a relative reduction in the amount of preference information elicited, but increases the interruption cost. This trade-off must be carefully considered when designing effective elicitation strategies.

In Chapter 6 we propose a probabilistic analysis of vote elicitation that combines the advantages of incremental elicitation schemes—namely, minimizing the amount of information revealed—with those of full information schemes—single (or few) rounds of elicitation. We exploit distributional models of preferences to derive the ideal *ranking threshold* k, or number of top candidates each voter should provide, to ensure that either a winning or a high quality candidate (as measured by max regret) can be found with high probability. Our main contribution is a general empirical methodology, which uses preference profile samples to determine the ideal ranking threshold for many common voting rules. We develop probably approximately correct (PAC) sample complexity results for one-round protocols with *any voting rule* and demonstrate the efficacy of our approach empirically on one-round protocols with Borda scoring.

1.2 Organization of Dissertation

In Chapter 2, we review the necessary background in social choice theory. This includes notation we will use throughout the dissertation. We will also review some related work on single-agent decision making with partial preference information, and prior work on elicitation in social choice. In Chapter 3 we present notions of robust decision assessment and regret-based elicitation in the context of social choice for voting schemes that selects a single alternative. We then go on to develop efficient and effective algorithms for robust decision making and elicitation for several common voting rules. In Chapter 4 we apply the robust social choice framework to the problem of selecting a subset or slate of alternatives. In particular, we develop robust decision support algorithms and an

effective elicitation heuristic for the proportional representation scheme, which is known to be NP-hard even with complete preference information. In Chapter 5, we present the first efficient algorithms for learning Mallows models with arbitrary collections of pairwise preference data. Chapter 6 presents a model of multi-round elicitation, and a probabilistic analysis of preference elicitation in a new one-round protocol that exploits populational ranking models. Finally Chapter 7 concludes the thesis and discuss future research directions.

Chapter 2

Overview of Social Choice

The field of social choice, and in particular, voting theory, is concerned with the problem of selecting an *outcome*, from a set of outcomes or *alternatives* (*candidates*, *items* are also terms used throughout this dissertation) $A = \{a_1, \ldots, a_m\}$, for a set of agents $N = \{1, \ldots, n\}$ (see Gaertner [52] for an overview). Such an outcome is a decision for all agents, that impacts each agent's satisfaction or utility. Agents, however, may have different preferences for different alternatives, and as such, the outcome must be chosen so that it reflects some notion of consensus, compromise, or fairness.

This can be achieved through two avenues, first through defining axioms that any selection, or *voting rule* should satisfy, and second through the principle of decision theory, where the goal is to maximize some objective or societal welfare function. The focus of this dissertation is mostly the latter approach. There are many excellent overviews of social choice and voting to which we refer the interested reader for a more detailed background, including general overviews [52, 97] as well as material with a more specific computational focus [16, 27, 31]. To begin, we present formal definitions of preference relations, partial preferences and other concepts central to computational social choice.

2.1 Preliminaries

2.1.1 Preference Relations

A *preference relation* R over a set A of alternatives is an anti-symmetric, transitive, binary relation such that for any $x, y \in A$ with $x \neq y$, either xRy or yRx. In other words R is a strict "prefers to" relation. We refer to R as a *complete preference* or *complete vote* or simply *ranking*, since alternatives can be linearly ordered with respect to R. It is common to use symbols $\succ$ for "is preferred to" and $\prec$ for "is dispreferred

to." It is possible to allow for weak preference relations, that is, relations that allow ties in preferences. We focus on strict preferences, however, most models and concepts presented in this dissertation can be readily extended for weak preferences. It is assumed each agent $\ell \in N$ has an underlying *complete* or *full* preference v_ℓ. Let $v_\ell(a)$ for $a \in A$ be the preference *rank* or *position* of a. We denote by 2^A the set of all subsets of A (also known as the *power set* of A). Let σ be a ranking, for any $X \in 2^A$ (i.e., $X \subseteq A$), let $\sigma|_X$ denote the ranking obtained by restricting σ to alternatives in X.

For a positive integer u let $[u] = \{1, \ldots, u\}$. As discussed above, R can be represented by a ranking, or equivalently a bijection $\sigma : A \to [m]$ which maps an item onto its *rank*. Thus for $i \in [m]$, $\sigma^{-1}(i)$ is the item in the ith rank. We will use the notation $\sigma = \sigma_1\sigma_2\cdots\sigma_m$ to refer to a ranking where the ith ranked item is σ_i. Let $\mathbf{1}[\cdot]$ be the indicator function.

The collection of agent preferences $\mathbf{v} = (v_1, \ldots, v_n)$ is referred to as a *preference profile*. Let $\mathbf{V}$ denote the set of all preference profiles, and $\mathbf{V}_{n,m}$ denote the set of all preference profiles for a fixed n and m. Note that $\mathbf{V}$ is not restricted—in this dissertation, we generally do not make assumptions about the structure of preference profiles, such as that of *single-peaked* preferences [10, 93].

2.1.2 Partial Preferences

While agents have a true underlying complete preference, in many applications only partial preferences are reported, elicited, or observed. It is possible to capture many incomplete ordinal preferences with a set of *pairwise preferences*

$$p = \{x_1 \succ y_1, \ldots, x_k \succ y_k\},$$

where we abuse the $\succ$ notation to allow for incomplete preferences. We will also write $x \succ_p y$ to indicate x is preferred to y in the *partial* (or *incomplete*) *preference* (or *vote*) p. We will use the term *preference* to refer to either a complete or partial preference as understood in context. Let $\mathsf{tc}(p)$ denote the *transitive closure* of p, which is the smallest transitive relation containing p. We write $\{x, y\} \in p$ if there is a comparison between x and y in p and, similarly $\{x, y\} \in \mathsf{tc}(p)$ if x and y are comparable under p's transitive closure. We assume p is *consistent*, that is, the transitive closure contains no cycles in p. Since preferences are strict, the transitive closure forms a *partial order* on A. Preference p is a full ranking if and only if $\mathsf{tc}(p)$ is a total order.

Denote by $\mathsf{C}(p)$ the set of *linear extensions* (or *completions*) of p, which is the set of all complete preferences that are consistent with p. Let $\mathsf{C} = \mathsf{C}(\emptyset)$ denote the set of all

$m!$ rankings. We will refer to the collection of incomplete preferences $\mathbf{p} = (p_1, \ldots, p_n)$, one for each voter, as a *partial preference profile*. We use $\mathsf{C}(\mathbf{p}) = \mathsf{C}(p_1) \times \cdots \times \mathsf{C}(p_n)$ to denote the set of *complete preference profiles* that are consistent with the partial votes.

There are several natural, special cases of pairwise partial preferences. For example, top-t preferences, in which agents rank their t most preferred alternatives, are quite popular, as well as its generalizations [79]. To illustrate, suppose we have a set of alternatives $A = \{a, b, c, d, e\}$. An example of a top-2 preference might be a partial ranking that ranks a first and b second, while not specifying the remaining part of the ranking with respect to c, d and e. This top-2 example corresponds to the following set of pairwise comparisons: $p = \{a \succ b, b \succ c, b \succ d, b \succ e\}$. Another interesting class of preferences are rankings of only subsets of items [58, 26]. For example, an individual may rank the subset $\{b, d, e\}$ in the following order: $e \succ b \succ d$. This corresponds to the following set of pairwise comparisons: $p = \{e \succ b, b \succ d\}$. The remaining alternatives a and c can be interleaved in any position within this ranking of b, d and e. However, all of these fail to capture natural partial preferences even as simple as exactly two pairwise comparisons: $p = \{a \succ b, c \succ b\}$. In this case, since we do not know the most preferred alternative (which may be any alternative except for b), we cannot represent p as a top-k preference structure. Furthermore, it is not a ranking of a subset of alternatives because we do not know the relative ranking of a and c. In this dissertation we treat pairwise comparisons as the fundamental building blocks of partial preference relations, since they represent almost all special cases of interest.

2.1.3 Distances over Preference Rankings

There are several notions of distance and metrics over rankings and partial preferences (see Diaconis [41] for an overview). We will focus on the popular Kendall-tau distance [72] and a simple extension to partial preferences. We will use the terms *disagreement* or *misordering* to refer to a pairwise comparison $a \succ b$ that differs from the preference $b \succ a$ in some ranking. Given a ranking $\sigma = \sigma_1 \sigma_2 \cdots \sigma_m$ over A and a partial preference p, define the *dissimilarity measure* $d(p, \sigma)$ to be the number of pairwise comparisons in the transitive closure $\mathsf{tc}(p)$ that are misordered with respect to σ. More formally,

$$d(p, \sigma) = \sum_{i<j\leq m} \mathbf{1}[\sigma_j \succ \sigma_i \in \mathsf{tc}(p)]. \tag{2.1}$$

Note that in preference ranking σ, alternative σ_i is preferred to σ_j whenever $i < j$. Hence, the indicator function within the above summation will count the number of

pairwise disagreement between $\mathsf{tc}(p)$ and σ. If p is a full ranking, then $d(p, \sigma)$ is the Kendall-tau distance. Effectively, d is an extension of the Kendall-tau distance where one argument can be a strict partial order. Likewise, define $s(p, \sigma)$ to be the number of pairwise comparisons in the transitive closure of p that agree with σ. That is,

$$s(p, \sigma) = \sum_{i<j \leq m} \mathbf{1}[\sigma_i \succ \sigma_j \in \mathsf{tc}(p)]. \tag{2.2}$$

Hence $d(p, \sigma) + s(p, \sigma)$ is the number of comparisons in the transitive closure of p. If p is a full ranking, then $d(p, \sigma) + s(p, \sigma) = \binom{m}{2}$.

2.2 Single-Choice Problems

The single-choice group decision problem [16, 27, 107], also known as the *winner determination* problem, is that of specifying and computing a winning consensus alternative given a preference profile. Before implementing or computing a winning alternative, one must specify a function that maps any possible preference profile into a single alternative.

Definition 1. Let N be a set of n agents and A be a set of m alternatives. A *social choice function (SCF)* is any mapping $f : \mathbf{V}_{n,m} \to A$.

For certain common voting rules, we assume the availability of a tie-breaking mechanism. If the tie-breaking is deterministic, we assume the SCF captures tie-breaking.

Definition 2. Given a social choice function f, and a preference profile $\mathbf{v} \in \mathbf{V}$, the *winner* is the alternative $f(\mathbf{v})$.

In this dissertation we assume SCFs are deterministic mappings. Stochastic SCFs may be useful as a protocol to break ties, for example, by selecting a winner at random amongst alternatives that have the highest number of first-place votes.

The Muller-Satterthwaite theorem [94] is an impossibility result for winner determination. It states that three seemingly desirable axioms cannot be simultaneously satisfied. We define these axioms as follows.

Definition 3 (Weak Unanimity)**.** A social choice function f is *weakly unanimous* if for all $\mathbf{v} \in \mathbf{V}$, and for all pairs $x, y \in A$ such that $x \succ_{v_\ell} y$ for each vote v_ℓ in $\mathbf{v}$, then $f(\mathbf{v}) \neq y$.

Weak unanimity simply asserts that if an alternative x is preferred to y by all agents, then the SCF should never select y.

Definition 4 (Monotonicity)**.** A social choice function f is *monotonic* if for all n, m, for all $\mathbf{v} \in \mathbf{V}_{n,m}$, if $f(\mathbf{v}) = x$, and $\mathbf{v}' \in \mathbf{V}_{n,m}$ is another profile such that $x \succ_{v'_\ell} y$ whenever $x \succ_{v_\ell} y$ for all agents ℓ and all alternatives y (where $v_\ell \in \mathbf{v}$ and $v'_\ell \in \mathbf{v}'$), then $f(\mathbf{v}') = x$.

The monotonicity condition essentially states that if one takes a preference profile $\mathbf{v}$ for which x is the winner, then modifying that profile by modifying each vote $v \in \mathbf{v}$ in such a way that the position of x might improve relative to other alternatives, then x remains the winner (i.e., $f(\mathbf{v}') = x$).

Definition 5 (Non-Dictatorial)**.** Let f be a social choice function. An agent $\ell \in N$ is a *dictator* (with respect to f) if for every $\mathbf{v} \in \mathbf{V}$, $f(\mathbf{v}) = v_\ell^{-1}(1)$. A social choice function f is *dictatorial* if there is a dictator and it is *non-dictatorial* if no dictator exists.

The last condition states that, f should not admit dictators—that is, an agent ℓ whose most preferred alternative, $v_\ell^{-1}(1)$, is always chosen as the winner. Finally, the Muller-Satterthwaite result states that the above three conditions cannot be simultaneously satisfied.

Theorem 1 (Muller-Satterthwaite [94])**.** *For any set of alternatives A such that $m \geq 3$, and there is no restriction on the set of preference profiles, if f is weakly unanimous and monotonic then it must be dictatorial.*

In many respects the Muller-Satterthwaite theorem resembles the famous impossibility result of Arrow [4], in the context of social choice functions (as opposed to *rank aggregation* or *social welfare functions* as considered by Arrow).

Another well-known impossibility result for social choice functions involves the *vote manipulation* phenomenon, when agents misreport their preferences to change the social choice outcome in their favor. The famous Gibbard-Satterthwaite theorem [54, 110] ensures that no sensible voting rule is resistant to manipulation when the manipulator(s) have full knowledge of other agents' true preferences.

Definition 6 (Unanimity)**.** A social choice function f is *unanimous* if for all $x \in A$, and all $\mathbf{v} \in \mathbf{V}$ such that for each $\ell \in N$, $v_\ell^{-1}(1) = x$, then $f(\mathbf{v}) = x$.

Unanimity asserts that for a preference profile where all agents rank the same alternative, x, as their most preferred, then f selects x.

Definition 7 (Manipulability)**.** A social choice function f is *manipulable* if there exists n, m, $\mathbf{v} \in \mathbf{V}_{n,m}$, an agent $\ell \in N$, and a different profile $\mathbf{v}' \in \mathbf{V}_{n,m}$ (where $v'_\ell \neq v_\ell$, $v'_{\ell'} = v_{\ell'}$ for all $\ell' \neq \ell$) such that $f(\mathbf{v}') \succ_{v_\ell} f(\mathbf{v})$. A *strategy-proof* social choice function f is not manipulable.

Manipulability asserts that there exist a preference profile and an agent who can change their vote in such a way that the winning outcome $f(\mathbf{v}')$ is preferred to $f(\mathbf{v})$ under the agent's original preference v_ℓ. The Gibbard-Satterthwaite result asserts the following:

Theorem 2 (Gibbard-Satterthwaite [54, 110])**.** *For any set of alternatives A such that $m \geq 3$, and no restriction on the set of preference profiles, if f is unanimous and non-dictatorial, then it must be manipulable.*

The above impossibility results, while of strong theoretical interest, have little bearing on the reality that many practical problems require the selection of a consensus choice. One desirable property of voting rules that Condorcet [30] proposed was that an alternative that is preferred to all other alternatives by a majority of voters should always win.

Definition 8 (Condorcet criterion)**.** Given a preference profile $\mathbf{v}$, an alternative $x \in A$ is a *Condorcet winner* if for all $y \in A$,

$$|\{\ell \in N : x \succ_{v_\ell} y\}| > n/2.$$

A social choice function f satisfies the *Condorcet criterion* if for all $\mathbf{v} \in \mathbf{V}$, $f(\mathbf{v}) = x$ whenever x is a Condorcet winner.

While this seems like a reasonable criterion it is turns out such an alternative may not exist. This is known as the *Condorcet paradox*.

Example 1 (Condorcet paradox)**.** Let $A = \{a, b, c\}$, $v_1 = abc$, $v_2 = bca$ and $v_3 = cab$. In the preference profile $\mathbf{v} = (v_1, v_2, v_3)$, none of the alternatives a, b or c is a Condorcet winner: a is only preferred to c once, b is only preferred to a once, and c is only preferred to b once.

However, rules that do elect the Condorcet winner—if it exists—are called *Condorcet methods* and include, for example, the rule that elects the top ranked alternative in a Kemeny consensus ranking (see Section 2.4). We will now review various popular voting rules for winner determination. This is only a brief review and covers the main aspects, including the technical definitions, of each voting rule.

Positional Scoring Rules

The first set of rules are known as *positional scoring rules*. Formally the *score* of alternative $a \in A$, given a *positional scoring function* (or *PSF*) $\alpha : [m] \to \mathbb{R}_{\geq 0}$ that maps rank

positions into a non-negative number, and a profile $\mathbf{v}$, is

$$s_\alpha(a, \mathbf{v}) = \sum_{\ell \in N} \alpha(v_\ell(a)).$$

We write $\alpha_\ell(a) = \alpha(v_\ell(a))$ which can be interpreted as a measure of ℓ's "satisfaction" with alternative a. The positional scoring winner, using PSF α, is the alternative with the highest score $a^* = \text{argmax}_{a \in A}\, s_\alpha(a, \mathbf{v})$. Special cases of positional scoring rules include Borda, Plurality, k-approval and k-veto.

The *Borda count* is attributed to Jean-Charles de Borda [39], a French mathematician and scientist whose proposed method was used by the French Academy of Sciences to elect its members. This rule has been used in a range of applications including political elections, university council elections, and awarding sports trophies. The rule is simple: a given ranking is translated into scores for the alternatives, where the alternative ranked ith receives a score of $m-i$. In other words it is a positional scoring rule with $\alpha(i) = m-i$ for $i \in [m]$.

Plurality is arguably the most commonly used voting rule in political elections. This is in part due to its ease of use (whereby voters need only report their most preferred candidate, as opposed to a full ranking) and its understandability. It is a special case of the positional scoring rule with $\alpha(1) = 1$ and $\alpha(i) = 0$ for $i > 1$.

k-Approval is a voting rule where each agent reports a k-set of alternatives, with the winner being the alternative that appears in the most agents' chosen approval set. For example, plurality is 1-Approval. It is a special class of positional rules where $\alpha(i) = 1$ if $i \leq k$, and $\alpha(i) = 0$ otherwise.

k-Veto is similar to k-Approval, where each agent reports a k-set of alternatives they *do not prefer*. It is equivalent to $(m-k)$-Approval where the approval set is the complement of the veto set. In particular, $\alpha(i) = 1$ if $i \leq m-k$ and $\alpha(i) = 0$ otherwise.

Approval Voting

A rule that is similar to positional scoring is *approval voting*, where agents submit a subset of alternatives, of *any size*, that they "approve." The winner is the alternative with greatest number of approvals across all agents. Approval voting is a simple method that is in popular use for a variety of applications including, for example, finding a common time for meetings.

Single Transferable Vote (STV)

Also known as *instant run-off*, STV is a reasonably common method for political elections in the English speaking world (e.g., Irish and Indian presidential elections, mayoral elections in London, Australian House of Representatives, etc.). Voters submit a top-k list of preferred items (i.e., a ranking of k most preferred alternatives). Winner determination proceeds in rounds, and in each round, if there is an alternative that has a majority ($\geq 50\%$) of the first place votes then it is elected, otherwise the alternative with the fewest number of first place votes is eliminated and the alternative is deleted from all the votes (i.e., in each agent's top-k ranking, the alternative is removed and every candidate ranked below is moved up). This process is repeated until a winner is found. There are also variations of this rule such as eliminating more than one candidate in each round.

Maximin

Maximin voting is a Condorcet method, defined as follows. Let

$$N(a_i, a_j; \mathbf{v}) = |\{v_\ell : v_\ell(a_i) < v_\ell(a_j), \ell \in N\}|,$$

be the number of voters who prefer a_i to a_j. Let

$$s_m(a_i, \mathbf{v}) = \min_{j \neq i} N(a_i, a_j; \mathbf{v}).$$

The alternative a with highest score $s_m(a, \mathbf{v})$ wins.

Copeland

The Copeland voting rule is also a Condorcet method, defined as follows. Let $W(a_i, a_j; \mathbf{v}) = 1$ if strictly more voters prefer a_i to a_j, $1/2$ if tied and 0 otherwise. Let

$$s_c(a_i) = \sum_{j \neq i} W(a_i, a_j; \mathbf{v}).$$

The alternative a with largest $s_c(a)$ wins.

Egalitarian (maxmin fairness)

The Egalitarian voting rule is defined by

$$s_f(a, \mathbf{v}) = \min\{m - v_\ell(a) : \ell \in N\}.$$

The winner is the alternative a that maximizes $s_f(a, \mathbf{v})$. This objective maximizes the satisfaction/utility of the least satisfied agent. One can extend this to incorporate arbitrary positional scores. While changing the positional scores does not change the winner, it may change the social welfare (i.e., the total score) of the winner.

Bucklin

The Bucklin rule is defined using the Bucklin score $s_B(a, \mathbf{v})$, which is the smallest $k \in \{1, \ldots, m\}$ such that more than half of all voters rank a above position k. The winner is the alternative a with smallest Bucklin score.

Range Voting

Smith [113] introduces the notion of range voting. The voting rule is similar to positional scoring, with the exception that rather than reporting rankings of alternatives, each agent gives a score for each alternative, with the score drawn from some predetermined fixed interval of values. Then each alternative's score is tallied across the votes and the alternative having the highest score wins. While this requires that agents specify "utility" values for each alternative (imposing significantly more cognitive burden than specifying a ranking), it allows for more flexibility, expressiveness, and strength of preference over Borda or positional scoring, and can thus lead to a better consensus winner. Approval voting is a special case of range voting where each voter chooses a value from $\{0, 1\}$ corresponding to approval or disapproval, for each alternative.

Cumulative Voting

Cumulative voting is similar to range voting except the sum of the scores submitted for each voter are normalized (e.g., to 1). This requires the voter to strategically distribute a fixed unit of scores across the alternatives.

2.3 Multi-Choice Problems

The *multi-winner election* problem is that of selecting a subset of the alternatives, usually of some restricted size K.

Definition 9. Let N be a set of n agents and A a set of m alternatives. A *multi-choice function (MCF)* is any mapping $f : \mathbf{V}_{n,m} \to 2^A$.

We will also refer to a MCF as a *multi-winner rule*. Multi-winner rules are typically motivated by the selection of committees, such as choosing members of parliament, boards of directors, etc. One can always frame this as a winner determination problem where the outcome corresponds to the admissible subsets of alternatives of the multi-winner problem. For example, each outcome in a multi-choice winner determination problem corresponds to a K-subset of alternatives, resulting in $\binom{m}{K}$ total outcomes. However, this is a large choice space, and imposes the unrealistic requirement of having agents rank all such subsets. Instead methods based on rankings over the original alternatives (rather than sets of alternatives) are more desirable.

One ad hoc approach to the problem is to use a rank aggregation rule such as Borda and pick the top K alternatives as the winning set. However, this heuristic may be inappropriate given the true objectives of the problem. Meir et al. [91] study problem settings, such as those using k-approval and approval votes, where such a heuristic (or variations of it) yield optimal solutions. They also investigate the computational complexity of manipulability. This heuristic is discussed in more depth in in Lu and Boutilier [83].

For the remainder of this section we instead focus on schemes that aim for *proportional representation*, which is the main motivation behind many multi-winner methods. The rough idea is to select a committee where the weight or number of elected candidates of a particular political party roughly corresponds to the number of voters that support that party. There are several rules that aim to implement proportional representation, which we outline below.

Single Transferable Vote (STV)

While STV allows one to select a winner, STV can be extended to selecting a set of winners is to run a number of rounds, each time either electing a winner with more than half of all first place votes or eliminating the alternative with fewest votes, until a set of K winners is obtained.

Chamberlin and Courant's rule

Chamberlin and Courant [25] propose a rule whereby a given K-subset $\overline{a} \subseteq A$ has an objective notion of quality given by

$$s(\overline{a}, \mathbf{v}) = \sum_{\ell \in N} \max_{a \in \overline{a}} s(a, v_\ell)$$

where $s(a, v_\ell) = m - v_\ell(a)$ is the Borda score. The winning set $\overline{a}^*$ maximizes $s(\overline{a}, \mathbf{v})$. One can extend this rule to arbitrary positional scores or utility values. Each voter's preference over a slates of alternatives is the preference of the alternative from which the voter benefits the *most*.

This is what Chamberlin and Courant term the "representation axiom." They provide political justifications of this idea. Roughly speaking, they argue that a voter is really interested in having one "representative," i.e., his/her most preferred candidate in the committee, to "speak" for the voter's interests. Having more candidates speak does not add to the voter's satisfaction. They also suggest, in passing, the possibility of trading off social welfare and the cost of larger committees: "examination of the rate at which the measure of representation increases as committee size increases would allow proper balance to be struck between the benefits of increased representation and the costs associated with increases in committee size." This is a trade-off that would occur in applications other than political voting, such as market segmentation. However, Chamberlin and Courant [25] do not discuss the problem of algorithmically finding the optimal committee. Potthoff and Brams [102] generalize Chamberlin and Courant's model to allow each agent to be represented by several candidates in the slate of winners. They also provide an integer programming formulation for optimal slate selection, shown below with slight modifications.

Let $x_i \in \{0, 1\}$, $i \leq m$ denote whether alternative a_i appears in the slate set $\overline{a}$, and let $y_{\ell i} \in \{0, 1\}$, $\ell \leq n, i \leq m$ denote whether a_i is agent ℓ's most preferred element in ba. We then have:

$$\max_{x_i, y_{\ell i}} \quad \sum_{\ell \in N} \sum_{i=1}^{m} \alpha_\ell(a_i) \cdot y_{\ell i} \tag{2.3}$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_i \leq K, \tag{2.4}$$

$$y_{\ell i} \leq x_i, \qquad \forall \ell \leq n, i \leq m \tag{2.5}$$

$$\sum_{i=1}^{m} y_{\ell i} = 1, \qquad \forall \ell \leq n. \tag{2.6}$$

More complexity and algorithmic results are provided by Lu and Boutilier [83].

The follow up work by Monroe [92] presents a new model termed "fully proportional representation" and is essentially the Chamberlin and Courant model, except that there is an explicit constraint requiring that the number of voters whose most preferred alternative is $x \in \overline{a}$ is roughly the same for every $y \in S$. In other words if voters are clustered by their favourite candidate in the committee, then the clusters are of roughly equal size.

Procaccia et al. [103] show that computing Chamberlin and Courant's rule as well as Monroe's rule, for approval votes (and not ranking votes), is NP-hard. See [112, 86] for additional computational results on this problem. Recently, Aziz et al. [6] studied the axiomatic properties of approval-based committee voting. Facility location [59] and maximum coverage problems [28] all bear close connection to these problems as well. We are unaware of any work that considers either robust optimization of slates in multi-winner problems in the presence of an incomplete preference profile, nor any work that considers the incremental elicitation of voter preferences in such a setting.

Budgeted Social Choice

Budgeted Social Choice was introduced by Lu and Boutilier [83], and is similar to Chamberlin and Courant's model but is considerably more general. The authors also provide new complexity and algorithmic results (including a greedy algorithm that has a $1 - \frac{1}{e}$ approximation ratio) as well as evaluating the solution quality of taking the top-k candidates in a Borda ranking (i.e., a ranking of candidates by their Borda counts).

Related is the combinatorial public project problem [98] where each agent has a valuation function over all subsets of alternatives, and a limited number of alternatives must be chosen for all agents. The focus is more on the tension between approximating social welfare and incentivizing truthfulness (which would require payments from agents).

Segmentation Problems

Kleinberg et al. [74] proposed a class of so-called *segmentation problems* that is very similar to Chamberlin and Courant's model. Specifically, it is assumed that each agent has a utility function $f_\ell(a) = w_\ell \cdot a$, the dot-product of a weight vector w_ℓ with a multi-attributed representation of alternative a. Then the segmentation problem involves finding K alternatives $S \subseteq A$ such that $\sum_{\ell \in N} \max_{a \in S} f_\ell(a)$ is maximized. This is very much like Chamberlin and Courant's model, and in fact is the same if alternatives lie in m dimensional space and a_i is represented by the vector with a 1 in the ith entry and 0 elsewhere; then w_ℓ is the vector of utility values for alternatives (i.e., w_ℓ represents Borda

scores). The authors show that it is NP-hard to optimize (specifically it is MAXSNP-complete) but that polytime algorithms exist for fixed dimension (i.e., fixed number of attributes over alternatives). They point out the trade-off between pure personalization (K large, which would be more costly as many alternatives must be procured) versus pure consensus (K small, which would be less costly). Often a balance must be struck: variable segmentation is defined by penalizing the objective function when K gets larger by a linear term $-\gamma K$ for some $\gamma > 0$. They also propose related problems and corresponding approximation algorithms.

2.4 Rank Aggregation

Rank (or *preference*) *aggregation* is one of the first problems that spurred much of the modern research into social choice. As opposed to single- or multi-winner voting rules, rank aggregation aims to combine multiple rankings into a consensus ranking.

Definition 10. A *rank aggregation rule* (also known as a *social welfare function* [4]) is any mapping $f : \mathbf{V} \to \mathsf{C}$.

Typically a rank aggregation rule outputs a consensus ranking that reflects the level of group preference over the alternatives. The rigorous study of rank aggregation schemes first started with the seminal work of Arrow [4]. Arrow's impossibility result spawned off much of the existing economic social choice literature. He was primarily interested in knowing whether there exists a rank aggregation rule that satisfies the following three seemingly desirable properties.

Definition 11 (Pareto Efficiency)**.** A rank aggregation rule f is *Pareto efficient* if for all $\mathbf{v} \in \mathbf{V}$, for all pairs $x, y \in A$, if $x \succ_{v_\ell} y$ for all $v_\ell \in \mathbf{v}$ then $x \succ_{f(\mathbf{v})} y$.

Pareto efficiency asserts that if all agents prefer x to y then the resulting aggregate ranking should rank x above y.

Definition 12 (Non-Dictatorial)**.** Let f be a rank aggregation rule. An agent $\ell \in N$ is a *dictator* (with respect to f) if for every $\mathbf{v} \in \mathbf{V}$, $f(\mathbf{v}) = v_\ell$. A rank aggregation rule f is *dictatorial* if there is a dictator, and it is *non-dictatorial* is no dictator exists.

In other words, the rank aggregation rule should not always output the ranking of the same agent.

Definition 13 (Independence of Irrelevant Alternatives (IIA))**.** A rank aggregation rule f is *independent of irrelevant alternatives (IIA)* if for all n, m, $\mathbf{v} \in \mathbf{V}_{n,m}$, for all pairs

$x, y \in A$ if $x \succ_{f(\mathbf{v})} y$, then for any $\mathbf{v}' \in \mathbf{V}_{n,m}$ such that $x \succ_{v_\ell} y$ if and only if $x \succ_{v'_\ell} y$, for all $\ell \in N$ ($v_\ell \in \mathbf{v}$ and $v'_\ell \in \mathbf{v}'$), then $x \succ_{f(\mathbf{v}')} y$.

That is, the relative ranking of two alternatives x and y, in the aggregate ranking, should depend on their relative rankings within the agents' preferences. Arrow's theorem asserts that the above three conditions cannot be simultaneously satisfied.

Theorem 3 (Arrow [4])**.** *For any set of alternatives A such that $m \geq 3$, if f is Pareto efficient and independent of irrelevant alternatives, then it must be dictatorial.*

Relatively recent work by Lu and Boutilier [82] casts questions about the utility of an aggregate ranking and provides a novel decision-theoretic model that not only justifies the use of a consensus ranking, but also motivates a well defined objective for rank aggregation.

While negative results are informative and provide us with an understanding of the difficulty inherent in selecting the right aggregation rule, such results do not suggest practical guidelines for aggregating rankings. For example, rank aggregation is of practical importance in the computational social choice, machine learning and information retrieval communities. It has found a prominent application in the so called "learning to rank" problem, which is vital in web search. We now discuss several of popular methods researchers have proposed for rank aggregation.

Positional Scores Ranking

To produce an aggregate ranking, one can use the positional scores of all alternatives (with respect to some PSF α), $s_\alpha(a, \mathbf{v})$, and rank them by their score in descending order. Ranking using the Borda score has been used in various rank aggregation applications [117, 45]. It is also known that sorting by Borda score yields a 5-approximation to the Kemeny ranking [38]. This provides a fast heuristic to the NP-hard problem of computing the Kemeny ranking.

Kemeny Ranking

The *Kemeny ranking* or *consensus* [71] represents a different approach to rank aggregation, which is based on pairwise comparisons instead of explicitly scoring alternatives. It is based on the Kendall-tau distance and outputs a ranking that minimizes

$$\kappa(r, \mathbf{v}) = \sum_{\ell \in N} d(r, v_\ell).$$

There are many interpretations of the Kemeny ranking, we will provide a more comprehensive and integrated discussion in Section 5.2.2.1.

Considerable work in computational social choice has focused on Kemeny. It is known to be NP-hard to optimize [8], even with four agents [45]. The optimization can be framed graph-theoretically: construct a *pairwise cost graph* where vertices represent alternatives and the weight of each directed edge $x \to y$ is $c_{xy} = |\{\ell \in N : y \succ_{v_\ell} x\}|$, the number of agents that prefer y to x. This is the "Kemeny cost" of choosing to place x above y in a consensus ranking. So the goal is to find an ordering r of the vertices $r = r_1 \dots r_m$ such that $\sum_{i<j} c_{r_i r_j} = \kappa(r, \mathbf{v})$ is minimized. Thus, the input to the problem of finding a Kemeny ranking does not require $O(nm)$ space to represent all rankings but simply $O(m^2)$ space to represent the pairwise cost graph.

One can formulate the optimization as an integer program in a straightforward way (see also [35]). Let $I_{xy} \in \{0, 1\}$ indicate whether $x \succ y$ is in the consensus ranking. We need transitivity constraints on the consensus ranking: $I_{xy} + I_{yz} - 2I_{xz} \leq 1$ for all distinct $x, y, z \in A$ as well as the anti-symmetric property $I_{xy} + I_{yx} = 1$ for all distinct $x, y \in A$. Then the objective function becomes

$$\min_{\mathbf{I}} \sum_{\{x,y\} \in A} I_{xy} c_{xy} + I_{yx} c_{yx}.$$

Despite the computational difficulty of constructing the Kemeny ranking, practical local optimization heuristics often work effectively [45]. This typically involves swapping a pair of items (local moves) in the candidate consensus ranking if it decreases the objective function until no more such pairs can be found. See Algorithm 1 for a local search heuristic. On the more theoretical side, a polynomial time approximation scheme exists [73] as well as approximation algorithms for nuanced extensions of Kemeny to partial rankings [2].

Other common rank aggregation rules include the Slater rule (which is similar to Kemeny but $c_{xy} = 0$ if majority of voters prefer x to y and $c_{xy} = 1$ otherwise), and the Dodgson rule. However, positional rules and Kemeny are most commonly used in practice.

2.5 Social Choice with Partial Preferences

Most of the social choice schemes discussed so far require each voter to provide a ranking of all alternatives. As discussed in Chapter 1, this can impose significant cognitive and communication burdens on voters, especially when the number of alternatives is large.

Algorithm 1 LocalKemeny

Input: Pairwise cost graph c_{xy} for all $\{x, y\} \in A$.

Initialize a ranking σ
$d \leftarrow \sum_{\{x,y\} \,:\, x \succ_\sigma y} c_{xy}$
for $i = 2..m$ **do**
  $x \leftarrow$ item in ith rank of σ
  for $j = i - 1..1$ **do**
    $y \leftarrow$ item in jth rank of σ
    if $c_{xy} < c_{yx}$ **then**
      Swap x with y
      $d \leftarrow d - c_{xy} + c_{yx}$
    else
      quit this loop
    end if
  end for
end for

Output: σ, Kemeny cost d

One prominent approach to reducing this burden is the development of mechanisms that guide the selection of winner(s) with partial votes, and corresponding elicitation strategies that cleverly query voters for the partial preferences that will best guide the selection of winner(s). The rest of this section will review previous work on solution concepts for social choice with partial votes, as well as results concerning elicitation problems for common voting rules.

2.5.1 Possible and Necessary Winners

One may be required to make a group decision without full preference information from agents, and to do so by being as "faithful" to the voting rule as possible. In such a situation, a number of approaches may be taken, which are described in detail in the following chapters. The predominant approach in computational social choice has been to rely on the notions of *possible* and *necessary* winners, as originally introduced by Konczak and Lang [75]. Roughly speaking, these concepts aim to prune the set of alternatives so that each remaining alternative either must win (necessary winners) or may win under *some* consistent completion (possible winners). Other approaches rely on probabilistic inference, choosing winner(s) assuming some underlying distribution of partial preferences.

Definition 14 (Possible Winner)**.** Let f be a social choice function, and $\mathbf{p}$ a partial preference profile. An alternative x is a *possible winner* if there exists a (full) preference

profile $\mathbf{v} \in \mathsf{C}(\mathbf{p})$ such that $x = f(\mathbf{v})$. Let $PW(\mathbf{p})$ denote the set of all possible winners with respect to $\mathbf{p}$ and f.

Definition 15 (Necessary Winner)**.** Let f be a social choice function, and $\mathbf{p}$ a partial preference profile. An alternative x is a *necessary winner* if for all $\mathbf{v} \in \mathsf{C}(\mathbf{p})$, we have $x = f(\mathbf{v})$. Let $f : \mathbf{V} \to 2^A$ be a social choice function that produces co-winners (does not break ties), then alternative x is a *necessary co-winner* if for all $\mathbf{v} \in \mathsf{C}(\mathbf{p})$, we have $x \in f(\mathbf{v})$. Let $NW(\mathbf{p})$ denote the set of necessary (co-)winners with respect to $\mathbf{p}$ and f.

We omit f from $PW(\mathbf{p})$ and $NW(\mathbf{p})$ to de-clutter our notation. In this dissertation the social choice function used when referring to possible and necessary (co-)winners will be clear from context. Note that the notion of a necessary co-winner is typically used in the context of single-choice voting rules. Examples include positional scoring rules as applied to preference profiles where two or more candidates have the maximum score. A somewhat similar, but not as strong definition as *non*-possible winners was studied earlier by Conitzer and Sandholm [32] in the context of their `ELICITATION-NOT-DONE` problem: given some complete votes with t votes still unknown (i.e., empty votes), is there a way to cast the t votes so a given candidate will lose?

It can be seen that $NW(\mathbf{p}) \subseteq PW(\mathbf{p})$ and, in particular, when $\mathbf{p}$ consists of all empty votes then $NW(\mathbf{p}) = \emptyset$ and $PW(\mathbf{p}) = A$, for many common voting rules. When enough preference information is known, $NW(\mathbf{p}) = PW(\mathbf{p})$, and no more preference information needs be obtained, even if the agent preferences are still partial. For example, when all agents state that x is their most preferred alternative, then clearly for the Borda rule $PW(\mathbf{p}) = NW(\mathbf{p}) = \{x\}$ and no more information is required from the agents. This highlights the fact that one may require much less preference information than that contained in full rankings to determine the winner(s).

Konczak and Lang [75] also show an interesting relationship between possible and necessary winners and *coalitional vote manipulation*. This problem considers a group of agents who wish to manipulate an election by either trying to elect a desired candidate (known as *constructive manipulation*) or trying to make a specific candidate lose (known as *destructive manipulation*) while having complete knowledge of other agents' preferences. These two computational problems can be framed as special cases of possible and necessary winners. In particular, by setting the manipulators' votes to all empty votes and the remaining agents' votes to their sincere votes, then asking if a desired candidate x is a possible winner is equivalent to asking whether the manipulators can vote in such a way as to make x win; similarly asking if a candidate is not a necessary winner is equivalent to asking whether there exists a destructive manipulation (see also footnote 3 of [32]).

In a follow up paper, Pini et al. [99] settle the computational complexity of possible and necessary winners for Single Transferable Vote (STV), where it is shown that finding possible and necessary (co-)winners is NP-complete and co-NP-complete, respectively. The NP-completeness result for STV possible winners essentially follows from Bartholdi III et al. [8], who showed the NP-completeness of deciding whether a particular voter, when given all other votes, can vote in such a way so as to make a given candidate win. Conitzer and Sandholm [32] also showed that the necessary winner problem for STV (even when only one voter's vote is unknown) is NP-complete. Note that it is easy to confirm [75] that the possible winners problems is in NP (for voting rules that are polynomial time computable) since a "certificate" consists of a valid complete profile and verifying the candidate winner in question would take polynomial time. The necessary winner problem is in co-NP since a certificate for a negative instance consists of a preference profile in which the candidate in question loses.

For specific rules, it has been shown by Xia and Conitzer [123] that computing possible winners is NP-complete for positional scoring rules (with respect to specific scores, such as the Borda count), Copeland, maximin, Bucklin, and ranked pairs. They also show necessary (co-)winners can be computed in polynomial time for positional scoring, maximin, and Bucklin, while showing it to be co-NP-complete for Copeland and ranked pairs. Konczak and Lang [75] show that possible and necessary (co-)winners can be computed in polynomial time for determining Condorcet winners. Pini et al. [99] show that for polynomial time computable rank aggregation rules which satisfy independence of irrelevant attributes (IIA), computing possible and necessary (co-)winners can be done in polynomial time (they define winner as the highest ranking candidate).

An open research direction proposed by Konczak and Lang [75] as well as Pini et al. [99] is whether there is a solution concept that would result in a smaller set of "good candidates" relative to a typically large set of possible winners while never being trivially empty. One approach is to count the frequency of winners in complete preference extensions, which gives a more probabilistic interpretation (assuming that all extensions are equally likley). Bachrach et al. [7] show the #P-hardness of this (essentially reducing from the problem of counting the number of linear extensions of partial orders, which is known to be #P-complete [20]). They also provide a randomized, high-level algorithm that can use, in principle, previously developed MCMC algorithms for sampling from the uniform distribution over consistent extensions of partial votes to approximately count the number of preference profiles in which a given alternative wins.

While much of the previous work discussed above has focused on possible and necessary winners, these concepts do not in fact directly support the selection or recom-

mendation of a particular group decision. The set of possible winners may be relatively large even with a considerable amount of preference information, and necessary winners may not exist unless the right kinds of preference information are known. Furthermore, one cannot quantify, and therefore distinguish, the quality of the alternatives within the possible winners. Chapter 3 delves into detail the weaknesses behind these concepts as a decision criterion.

2.5.2 Elicitation

Conitzer and Sandholm [32] first studied the problem of preference elicitation in the context of voting systems. They defined the notion of *coarse elicitation* as successively querying each agent for their full ranking and *fine elicitation* as querying for finer grained preferences, such as a single pairwise comparison or a top-k ranking. They introduced the `EFFECTIVE-ELICITATION` problem for coarse elicitation: given a complete preference profile and an integer $k > 0$, does there exist a subset of votes of size at most k such that those votes completely determine the winner(s)? In other words, regardless of what the other votes are, do the k elicited votes guarantee that the necessary co-winners are the same as the possible winners? The motivation is to understand whether, given the complete preference profile in hindsight, the optimal elicitation strategy can be found in polynomial time (optimal in the sense that after eliciting k rankings the possible winners and necessary (co-)winners coincide). They show that this problem is NP-complete for approval, Borda, Copeland and maximin rules. They also consider the strategy-proofness of fine elicitation schemes and conclude that generally, such schemes open new opportunities for agents to manipulate but that it can be controlled by restricting the form of the elicitation strategy, which may reduce its effectiveness in minimizing preference queries. A follow-up paper by Walsh [121] provides further complexity results for other voting rules, such as Copeland, cup, and ranked pairs, showing that in most cases it is NP-hard to determine when to terminate elicitation (similar complexity results can also be obtained from the complexity of destructive manipulation problems [36]).

Conitzer and Sandholm [34] investigate the *communication complexity* of common voting rules. In the context of elicitation for voting, communication complexity [125] refers to the minimum number of bits, over all preference profiles of some fixed dimension (i.e., fixed n and m), that need to be transferred from the agents to the decision maker so that the decision maker is able to determine the winner. They showed that for Borda, Copeland, and ranked pairs, agents' must communicate $\Omega(nm \log m)$ bits. This is equivalent to the amount of information needed to specify a full preference profile, within

a constant factor. This also holds for other popular rules such as approval, plurality, and plurality with runoff. There are a few other rules such as STV and maximin where there exist elicitation schemes that asymptotically query for less preference information than that contained in full rankings, in the worst-case. Note that because communication complexity is measured over the worst possible preference profile, some preference profiles that do not require much communication (e.g., a profile where every agent has the same top ranked candidate) are not reflected in this measure. In fact, as these results are worst-case, they do not necessarily provide insights into the average amount of preference information required for most real-world instances of preference profiles.

Almost no prior work has addressed practical vote elicitation—with the exception of the work of Kalech et al. [69]. They propose practical elicitation procedures for common voting rules. They focus on range voting, the Borda rule and a few other rules. The solution concept used to drive elicitation is that of possible and necessary (co-)winners. In particular, whenever the set of possible winners is equal to the set of necessary (co-)winners then elicitation can terminate and the winner(s) can be determined. Because computing possible winners for the Borda rule is NP-hard[1], they redefine possible winners as the candidate whose maximum possible score is larger than the minimum possible scores of all other candidates. Strictly speaking, this is a different notion of possible winners, but their definition includes all possible winners (in the original sense) and potentially other alternatives that are not possible winners.

Kalech et al. propose two elicitation algorithms. The first is a very simple and straightforward strategy that in each round, asks each voter, in some predetermined order, what their next top ranked alternative is (in round 1, it would query for their most preferred alternative, in round 2 query for second most preferred alternative, etc.) Elicitation terminates when the set of possible winners (according to their definition) is equal to the set of necessary co-winners. Their second greedy elicitation strategy tries to find a good but not provably optimal candidate. This is done by using the first elicitation algorithm and restricting the number of rounds to an input integer $P > 0$, then using $Q > 0$ (Q an input) alternatives with the largest possible minimum scores, and querying all voters for each of the Q alternatives' rank positions (or utility values). Finally, elicitation outputs the Borda or range vote winner over the Q candidates. As the authors acknowledge, there's no guarantee on the optimality or quality of the selected candidate.

In this dissertation, we take a *decision-theoretic* approach to the selection of a win-

[1]One might argue that a drawback of the possible winners concept is that deciding when to terminate elicitation becomes NP-hard.

ner or slate of winners, in the sense that we seek to find and quantify the best, or approximately good outcome with respect to the underlying social welfare objective. In particular, we use the notion of *minimax regret* as our criterion for *robust decision support*. The following chapters will define minimax regret in the context of social choice problems.

Once we have defined such a solution concept and are able to compute it exactly or approximately, we must assess the quality of the decision and whether it is sufficient for the particular problem. If it is of insufficient quality, the decision maker may have the opportunity to query for more preference information from agents until the quality meets a predefined threshold (possibly including true optimality). Finding the right elicitation strategy that will query for the least amount of preference information until the decision is of sufficient quality, is one of the main algorithmic problems that we will address. Note that the design of elicitation strategies will depend critically on the choice of the underlying decision criterion, in our case, the minimax regret criterion. A successful strategy will always aim to generate as few queries as possible to satisfy the criterion.

There are variety of forms of preference queries one can use to elicit votes. Much of this depends on the voting rule being used, and hence, the type of preference required. For example, in plurality voting one only needs to query the most preferred alternative of each voter. In approval voting one may ask whether a voter approves of a certain candidate or whether they like candidate x better than y (if so, this might be interpreted as x being approved and y not).

In rank-based voting rules, perhaps the most fundamental preference query is the pairwise comparison. Another natural query is to ask agents to rank their k most preferred alternatives (top-k). This, however, can be represented by a set of pairwise comparisons. One can also query for the least preferred alternatives, and this again can also be represented by a set of pairwise comparisons. Choice set queries are also possible, which involves presenting an agent with a slate of choices and asking which one(s) she prefers the most. Choice set queries can also be translated into a set of pairwise comparisons.

One might also ask what range of rank positions a certain candidate belongs in. This resembles a bound query where an agent is asked for an interval of possible utility values. However, it does not directly translate into pairwise comparisons, although one can argue that if an agent knows candidate x belongs in positions 2 to 5 then that agent should know what candidate belong in position 1 and all positions below 5, in which case it becomes a *partitioned preference* (see Definition 18 in Chapter 5) which can be represented by a set of pairwise comparisons. A special case of this query form is to ask what candidate is in rank position r.

For numerical based voting schemes such as range or cumulative voting, pairwise preference queries are less informative for deciding the winner. Bound queries and queries of the form "what is your most preferred alternative and what is your score for it?" (as introduced in [69]) are more natural and informative.

Chapter 3

Robust Optimization and Elicitation for Single-Choice Problems

This chapter introduces the concept of robust social choice, and preference elicitation for making single-winner group decisions with arbitrary partial preferences from the agents. This is in contrast to previous notions of possible and necessary winners in that it gives worst-case guarantees on the quality of the recommended alternative. This chapter shows that, in practice, one only needs to elicit a fraction of the full preference information in order to make an optimal group decision. Furthermore, if an approximately optimal decision is sufficient for the application domain, even less preference information is required.

We start in Section 3.1 by presenting the regret-based criterion, known as *minimax regret*, that is at the heart of our decision-theoretic approach to robust decision support. In the following section, Section 3.2, we investigate the computation of minimax regret. We show computational complexity results and connections to necessary winners. We also develop polynomial time algorithms for various voting rules, including Borda, Maximin, Egalitarian, and Bucklin, based on the construction of worst-case completions of partial profiles. In Section 3.3, we develop preference elicitation algorithms that quickly reduce minimax regret and we demonstrate its empirical effectiveness in reducing information requirements on both real datasets and synthetic datasets in Section 3.4. In Section 3.5, we discuss related work and in particular describe previous approaches to winner determination with partial preferences. Finally we summarize and conclude in Section 3.6.

3.1 Robust Winner Determination

In this section, we address the question of *robust winner determination* given partial information about voter preferences. We introduce the notion of *minimax regret* as a robustness criterion for such problems in Section 3.1.1. We discuss the relationship of minimax regret with the notions of possible and necessary winners in Section 3.1.2, and use this relationship to show that computing minimax regret is computationally intractable for certain voting rules (e.g., Copeland, ranked pairs).

Notice that most of the voting schemes discussed so far explicitly *score* alternatives with respect to agent preferences, implicitly defining some form of "societal utility," "degree of societal acceptance" or aggregate quality measure for each alternative. Indeed, this is true of many (though not all, such as STV) voting schemes.

Definition 16. Let N be a set of n agents and A a set of m alternatives. A *scoring function* is any mapping $s : A \times \mathbf{V}_{n,m} \to \mathbb{R}$.

In the sequel, we assume the existence of a scoring function $s(a, \mathbf{v})$ that measures the quality of any candidate given a preference profile $\mathbf{v}$.

Definition 17. Let r be a social choice function and s a scoring function. Then s is *consistent with* r if for all $\mathbf{v} \in \mathbf{V}$, $r(\mathbf{v}) \in \text{argmax}_{a \in A}\, s(a, \mathbf{v})$.

This is, of course a minimal requirement, since any voting rule can be defined using an indicator function as the score (i.e., by giving the winning alternative a score of one and all other alternatives a score of zero) . However, most rules discussed so far have "natural" scoring functions. Our approach to robust optimization will exploit this fact. When there are ties among the highest scoring candidates, usually some form of tie-breaking is used. None of our results are tied critically to any specific form of tie-breaking.

3.1.1 Minimax Regret

Let r be a voting rule, defined using some natural scoring function $s(a, \mathbf{v})$ such that $r(\mathbf{v}) \in \text{argmax}_{a \in A}\, s(a, \mathbf{v})$ as described above. For example, in plurality voting, the natural score $s(a, \mathbf{v})$ of alternative a is the number of votes that rank a first. For positional scoring more generally (of which plurality, k-approval, and k-veto systems are all instances), the score of an alternative is its total positional score. The natural score for egalitarian (maxmin fairness) is $s_f(a, \mathbf{v}) = \min\{m - v_\ell(a) : \ell \in N\}$.

Suppose we have a partial profile $\mathbf{p}$ and we are forced to make a decision in the face of this incomplete information. We distinguish our *partial information* setting from the

question of aggregating preferences of voters whose preferences reflect genuine *incomparability* (see, e.g., [100]). Unfortunately, the notions of necessary and possible winners do not resolve this issue satisfactorily: necessary winners are not guaranteed to even exist given arbitrary partial profiles; and possible winners can only be used to narrow the set of options rather than prescribing an actual winner. Here we propose the use of the *minimax regret* solution concept. This concept has been used for robust decision making, and for driving preference elicitation, in a variety of single-agent domains [14, 13, 19] and in mechanism design [67]; but our work is the first application of the notion to voting and (rank-based) social choice.

Intuitively, we measure the quality of any proposed winner $a \in A$ given $\mathbf{p}$ by considering how far from optimal a could be in the worst-case (i.e., given any completion of $\mathbf{p}$). The minimax optimal solution is any alternative that is nearest to optimal in the worst-case. More formally, we define:

$$\begin{aligned} \mathit{Regret}(a, \mathbf{v}) &= \max_{a' \in A} s(a', \mathbf{v}) - s(a, \mathbf{v}) && (3.1)\\ &= s(r(\mathbf{v}), \mathbf{v}) - s(a, \mathbf{v}) \\ \mathit{PMR}(a, a', \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(a', \mathbf{v}) - s(a, \mathbf{v}) && (3.2)\\ \mathit{MR}(a, \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} \mathit{Regret}(a, \mathbf{v}) \\ &= \max_{a' \in A} \mathit{PMR}(a, a', \mathbf{p}) && (3.3)\\ \mathit{MMR}(\mathbf{p}) &= \min_{a \in A} \mathit{MR}(a, \mathbf{p}) && (3.4)\\ a^*_{\mathbf{p}} &\in \operatorname*{argmin}_{a \in A} \mathit{MR}(a, \mathbf{p}) && (3.5) \end{aligned}$$

$\mathit{Regret}(a, \mathbf{v})$ is the loss (or regret) of selecting a as a winner, given true vote profile $\mathbf{v}$, instead of choosing the optimal alternative under rule r (equivalently, under scoring function s).[1] $\mathit{PMR}(a, a', \mathbf{p})$ denotes the *pairwise max regret* of a relative to a' given partial profile $\mathbf{p}$. This is simply the worst-case loss—under all possible realizations of the full profile—of selecting alternative a rather than a'. Notice that pairwise max regret can be negative. $\mathit{MR}(a, \mathbf{p})$ is the maximum regret (or max regret) of a, in other words, the worst-case loss associated with selecting a rather that selecting a true (score-maximizing) winner. We can view this as *adversarial selection* of a complete profile $\mathbf{v}$ to maximize the loss between our chosen alternative a and the true winner under $\mathbf{v}$. Our aim is to choose the alternative a with *minimum max regret*: $\mathit{MMR}(\mathbf{p})$ denotes minimax regret

[1] See Smith [113] who uses score-based regret to measure the performance of various voting rules, including range voting.

under partial profile $\mathbf{p}$, while $a^*_{\mathbf{p}}$ denotes the minimax optimal alternative.[2] This gives us a form of robustness in the face of vote uncertainty: every alternative has worst-case error at least as great as that of $a^*_{\mathbf{p}}$.

3.1.2 Relationship to Possible and Necessary Winners

Notice that if $MMR(\mathbf{p}) = 0$, then the minimax winner $a^*_{\mathbf{p}}$ has the same score or utility as the winner in any completion $\mathbf{v} \in \mathsf{C}(\mathbf{p})$; i.e., $a^*_{\mathbf{p}}$ is guaranteed to be optimal. While this does not imply there is a necessary winner under $\mathbf{p}$ (due to tie-breaking), $MMR(\mathbf{p}) = 0$ if and only if there is a necessary co-winner. Thus for any rule r we have, by setting $\varepsilon = 0$:

Observation 1. The max regret decision problem for any voting rule r (i.e., does alternative a have $MR(a, \mathbf{p}) \leq \varepsilon$) is at least as computationally hard as the necessary co-winner problem for r.

This implies that computing minimax regret is co-NP-hard for the following voting rules [124]: Copeland, ranked pairs, (balanced or unbalanced) voting trees, and STV (single transferable vote). This observation does *not* imply, however, the easiness of either max regret or minimax regret computations when the necessary co-winner problem is easy; but we describe polynomial time algorithms to compute minimax regret for several important voting rules in Section 3.1.1.

The relationship between max/minimax regret and possible winners is more complicated. For certain scoring rules (e.g., plurality) the minimax winner $a^*_{\mathbf{p}}$ must be a possible winner under $\mathbf{p}$. However, in general, we have:

Observation 2. The regret-minimizing alternative may not be a possible winner for some voting rules and partial profiles.

Figure 3.1 illustrates this observation, showing a vote profile in which, under the 2-approval voting rule, the alternative that minimizes max regret is not a possible winner.[3] Both possible winners in this example have a poor 2-approval score under *some* completion of the votes, while a compromise candidate that cannot win under *any* completion has a much higher guaranteed score (i.e., lower max regret) than either possible winner. This suggests that using the notion of possible winners to select winners with partial votes can be problematic for some voting rules. Indeed, there would appear to be no general

[2] We informally write as if the optimal candidate is unique, but there can be several alternatives a that minimize max regret.

[3] In 2-approval, the top two candidates in each voter ranking each receive one point, and the winner is that with highest total approval score.

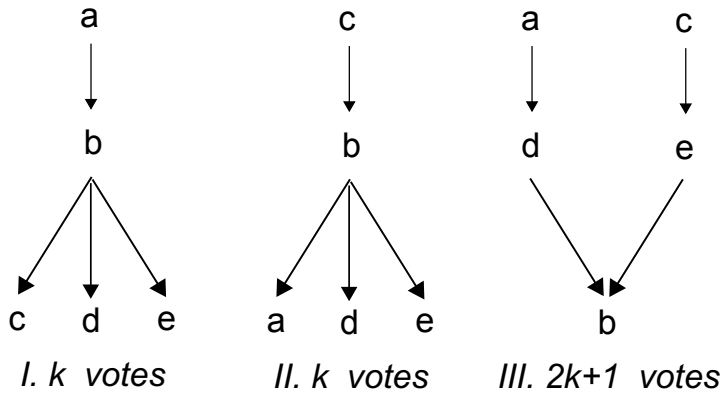

Figure 3.1: A partial profile $\mathbf{p}$ where the minimax alternative is not a possible winner (under 2-approval). Alternative b has score $2k$ in every completion. Either a or c must be at the top of every vote in set III, so one of them must receive at least $k+1$ approval points from set III. Hence $\max(s(a), s(c)) \geq 2k+1$, and a and c are both possible winners, while b is not. Now, $MR(b, \mathbf{p}) = k+1$ (a completion that puts a at the top of all votes in set III would give a a score of $3k+1$, the maximum possible). But $MR(a, \mathbf{p}) = 2k+1$: if we select a, the adversary will place c and e above a in each vote in set III, setting $s(a) = k$ and $s(c) = 3k+1$. $MR(c, \mathbf{p}) = 2k+1$ by similar reasoning.

way to ensure a possible winner isn't *far from being optimal* without using max regret to quantify this risk. The fact that the minimax winner $a^*_{\mathbf{p}}$ is not a possible winner is not problematic in our view, but if one insists on selecting from the set of possible winners, max regret could at least be used to aid in that selection (i.e., to choose the possible winner with least max regret). Still we take max regret to be the more fundamental notion for winner determination with partial information.

3.2 Computing Single-winner MMR

Minimax regret decision problems can often be solved as a *mixed integer program (MIP)* [14, 13] or a search problem [19] in a variety of decision problems. In our voting context, a MIP formulation (with variables capturing rank placement in specific votes) would be prohibitively expensive to solve. However, for certain voting rules and preference constraints, we can greatly simplify minimax regret computation by directly considering properties of the worst-case completions of voter profiles without directly computing them. We will illustrate this for several voting rules in this section. Our constructions are tightly related to those used by Xia and Conitzer [124] to demonstrate polynomial time algorithms for necessary winners for the positional scoring, maximin, and Bucklin rules. Indeed, their constructions can be viewed as attempting to maximize the difference in score between a proposed winner and an "adversarially chosen" alternative. We adapt

these ideas to minimax regret, and extend the analysis to maxmin fairness.

3.2.1 Exploiting Pairwise Max Regret

To demonstrate the polynomial time computability of minimax regret for specific voting rules below, we explicitly compute the pairwise max regret $PMR(a, w, \mathbf{p})$ of all $\frac{m(m-1)}{2}$ ordered pairs of alternatives (a, w) such that $a \neq w$ (where a is a proposed winner and w is an adversarial witness). With PMR in hand, we can readily determine minimax regret using Eqs. 3.3 and 3.4. We thus need only show that PMR can be computed in polynomial time.

A scoring function is *(additively) decomposable* if $s(a, \mathbf{v}) = \sum_{\ell \in N} s(a, v_\ell)$; i.e., if it is the sum of votewise scores. This implies that (pairwise) regret is decomposable, since

$$\begin{aligned} \mathit{Regret}(a, w, \mathbf{v}) &= s(w, \mathbf{v}) - s(a, \mathbf{v}) && (3.6)\\ &= \sum_{\ell \in N} s(w, v_\ell) - \sum_{\ell \in N} s(a, v_\ell) && (3.7)\\ &= \sum_{\ell \in N} [s(w, v_\ell) - s(a, v_\ell)]. && (3.8) \end{aligned}$$

Given a collection of partial votes $\mathbf{p}$, their completions by an adversary can be undertaken independently, so we can compute PMR by independently choosing the completions v_ℓ of each p_ℓ that maximize v_ℓ's local regret:

$$\begin{aligned} PMR(a, w, \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(w, \mathbf{v}) - s(a, \mathbf{v}) && (3.9)\\ &= \sum_{\ell \in N} \max_{v_\ell \in \mathsf{C}(p_\ell)} s(w, v_\ell) - s(a, v_\ell). && (3.10) \end{aligned}$$

All positional scoring functions are decomposable in this way.

3.2.2 Positional Scoring Rules

We illustrate our constructions by first examining the relatively simple case of computing $PMR(a, w, \mathbf{p})$ for a *linear* positional scoring rule.[4] Since PMR is decomposable, we determine, for any partial vote p, the completion v with maximum contribution to PMR.

For a given partial preference p, let $Nec(x \succ y)$ denote that x must be preferred to y in p, and $Inc(x, y)$ denote that no information is known regarding the pairwise preference of

[4]Linear means that an alternative's score is a linear function of its rank in v, hence the *difference* in two rank positions uniquely determines their difference in score. k-veto, k-approval, and plurality are not linear, but Borda is (linear rules are all "Borda-like").

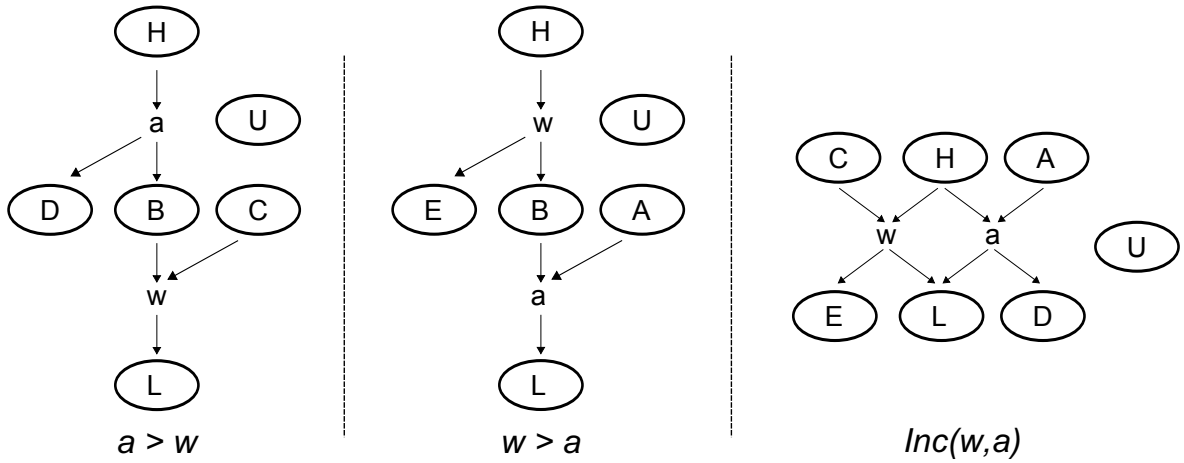

Figure 3.2: An illustration of the three possible relations between alternative a and adversarial alternative or "witness" w in any partial vote p. Other alternatives stand in a specific relation to a and w, with each oval representing one of the possible relationships (see text). To maximize a partial vote's contribution to pairwise max regret $PMR(a, w, \mathbf{p})$, linearizations of p require placing the groups of alternatives indicated by the ovals in specific positions relative to a and w in a way that depends on the scoring function.

x and y. Figure 3.2 illustrates the three different cases we need to consider in any partial vote p: (1) p implies $Nec(a \succ w)$; (2) p implies $Nec(w \succ a)$; and (3) p implies $Inc(w, a)$. In each case, the remaining alternatives stand in one of several possible relationships with a and w, as indicated by the ovals, each representing the set of alternatives that fall into the following categories:

- H: those alternatives that are (known to be) preferred to both a and w in the partial vote p.
- B: those alternatives that are between a and w in p. If $Nec(a \succ w)$ then these are preferred to w, while a is preferred to them. If $Nec(w \succ a)$ then these are preferred to a, while w is preferred to them. If $Inc(w, a)$, then there are no such alternatives.
- L: those alternatives that are dispreferred to both a and w.
- D: those alternatives that are dispreferred to a but incomparable to w. If $Nec(w \succ a)$, then there are no such alternatives.
- E: those alternatives that are dispreferred to w but incomparable to a. If $Nec(a \succ w)$, then there are no such alternatives.
- C: those alternatives that are preferred to w but incomparable to a. If $Nec(w \succ a)$, then there are no such alternatives.

- A: those alternatives that are preferred to a but incomparable to w. If $Nec(a \succ w)$, then there are no such alternatives.

- U: those alternatives that are incomparable to both a and w.

The lack of preference arrows between between certain sets (ovals) does not mean that pairwise preferences are not known between elements of some of these pairs. For instance, the preference $c \succ u$ for some $c \in C$ and $u \in U$ may be part of the partial vote. This relation will not play a role in constructing a completion for PMR, so is ignored. Of course, not all such pairwise preferences across distinct sets are viable—for instance, it is not possible to have $u \succ c$, since this would imply $u \succ w$, contradicting the definition of U. Similarly, elements within these sets may have known preferences within p, for instance, we may know $c \succ c'$ for some $c, c' \in C$. Indeed, we will exploit these "internal" known preferences in determining queries during preference elicitation (see Section 3.3). We will use these categories in analyzing other voting rules below as well.

Returning to linear positional scoring rules, in the first case, we have $Nec(a \succ w)$, so p's contribution to PMR must be negative. It is easy to see that we maximize pairwise regret with a completion v that minimizes the positional gap between a and w (i.e., maximize the adversary's (negative) advantage). To minimize the gap, it suffices to: (i) order set D below w, i.e., assume $w \succ d$ for all $d \in D$; (ii) order set C above a, i.e., assume $c \succ a$ for all $a \in A$; and (iii) order elements of set U either above a or below w. These orderings can be arbitrary (one need only maintain consistency within and across the sets in question). This implies that the (negative) contribution to PMR is exactly $-(|B| + 1)$. Note that we needn't compute an actual linearization of p, but simply need to determine the cardinality of the set B.

The second case of $Nec(w \succ a)$ proceeds similarly (see figure), but instead we *maximize* the positional gap between w and a (i.e., maximize the advantage of w over a). This is accomplished by ordering sets E, A and U (arbitrarily) between w and a. Hence, the contribution to PMR by p in the second case is $|B \cup F \cup E \cup U| + 1 = m - |H \cup L| - 1$. Finally, in the third case of $Inc(a, w)$, the (positive) advantage of w over a is maximized by ordering w over a and placing sets E, A and U between the two.

Computing PMR thus requires, for each partial vote, categorizing all alternatives as belonging to the relevant sets described above and indicated in in Figure 3.2. This is a simple matter: one compares each alternative a' to both a and w in the partial vote (assuming the transitive closure of p is given), classifying it into the appropriate set, which takes $O(m)$ time. This implies that $PMR(a, w, \mathbf{p})$ computable in $O(nm)$ time for linear scoring rules. With $O(m^2)$ pairs, computing $MMR(\mathbf{p})$ (and the optimal $a^*_{\mathbf{p}}$ and

its witness) takes $O(nm^3)$ time. In many practical settings, m can be treated as a small constant relative to n, in which case our algorithms scale linearly in n.

With linear positional scoring rules, arbitrary placement of alternatives that do not influence the positional gap between w and a (e.g., set U when $Nec(a \succ w)$) is allowed. For nonlinear rules, the size of the gap *and the position* of both a and w can influence w's advantage. However, the required placement can be found by simply examining splits of the set U of different cardinalities to determine how many to place above a and below w to minimize a's advantage over w—again, this can be accomplished in $O(m)$ time. Certain special cases can be treated more efficiently; e.g., if a positional rule is *monotonic non-increasing* (i.e., $s_i - s_{i+1} \geq s_{i+1} - s_{i+2}$) then U is placed above a (and if non-decreasing, below w). In any case, the minimax regret computation remains $O(nm^3)$. These observations show that:

Theorem 4. *Minimax regret can be computed in $O(nm^3)$ time for any positional scoring rule.*

3.2.3 Maximin Voting

We now consider pairwise max regret computation for the maximin voting rule, a non-decomposable voting rule that requires more intricate computation. However, it is "semi-decomposable" in the sense that we can compute independent completions of each partial vote p_ℓ, for each of a number of alternatives, and then aggregate the resulting scores.

To compute PMR, we recall the definition of the maximin scoring function, embedding it in the definition of PMR:

$$\begin{aligned} PMR(a, w, \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s_m(w, \mathbf{v}) - s_m(a, \mathbf{v}) \\ &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s_m(w, \mathbf{v}) - \min_{a' \neq a} N(a, a'; \mathbf{v}) \\ &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s_m(w, \mathbf{v}) + \max_{a' \neq a}(-N(a, a'; \mathbf{v})) \\ &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} \max_{a' \neq a} s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v}) \\ &= \max_{a' \neq a} \left[\max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v}) \right]. \end{aligned} \tag{3.11}$$

Let $M_{a'} = \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s_m(w, \mathbf{v}) - N(a, a'; \mathbf{v})$ denote the quantity inside the square brackets of Eq. 3.11. This represents the worst-case (over completions of $\mathbf{p}$) difference of the maximin score of w and the number of votes, $N(a, a'; \mathbf{v})$, in which a is preferred to a'.

We proceed by describing an algorithm for computing the pairwise max regret, $PMR(a, w, \mathbf{p})$,

which can then be used for computing and finding the minimax optimal alternative. Given a fixed a and w, we must consider all alternatives $a' \neq a$ in order to compute the expression in Eq. 3.11. Fix such an a', and now consider worst-case completions of each p_ℓ for this a':

- If $Nec_\ell(w \succ a)$ or $Inc_\ell(a, w)$, then we maximize the advantage of w over a by placing w as high as possible and a as low as possible in the worst-case completion of p_ℓ, using the techniques described above.
- If $Nec_\ell(a \succ w)$, consider all other alternatives b ($b \neq a$, $b \neq w$):
  - if $Nec_\ell(a \succ b)$ and $Inc_\ell(b, w)$, then place b below w;
  - if $Nec_\ell(b \succ w)$ and $Inc_\ell(b, a)$ then place b above a;
  - if $Inc_\ell(b, a)$ and $Inc_\ell(b, w)$, then: (1) place b below w if $b \neq a'$; or (2) place b above a otherwise;
  - otherwise place b arbitrarily, subject to the partial preference constraints.

Let $\mathbf{v}'$ be the completion of the partial votes p_ℓ in $\mathbf{p}$ as specified above. As above, specific completions need not be constructed, as the computations below can be largely performed using the cardinality of "sets" of alternatives relating w, a and a'. We compute $M_{a'} = s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}')$ for each a', and take the largest such $M_{a'}$ to be the pairwise max regret.

We now argue that this algorithm computes PMR correctly. Assume that we have a worst-case completion $\mathbf{v}^* \in \mathsf{C}(\mathbf{p})$ that is a maximizer of $M_{a'}$ (i.e., $s_m(w, \mathbf{v}^*) - N(a, a'; \mathbf{v}^*) = M_{a'}$). We claim that we can reconfigure $\mathbf{v}^*$ into another completion $\mathbf{v}'$ of $\mathbf{p}$ (exactly as described above) such that $s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}') = M_{a'}$. For any v_i^* and its corresponding partial vote p_ℓ, the reconfiguration is the same as for positional scoring rules in the cases when $Nec_\ell(w \succ a)$ or $Inc_\ell(a, w)$—that is, we place w as high as possible and a as low as possible in the resulting ranking.

The case $Nec_\ell(a \succ w)$ is different. One must place alternatives b where $Nec_\ell(a \succ b)$ and $Inc_\ell(b, w)$ below w. This does not change $N(a, a'; \{v_i^*\})$, and does not decrease the maximin score of w, even if $b = a'$. If b is such that $Nec_\ell(b \succ w)$ and $Inc_\ell(b, a)$, then we place b above a: this does not change the maximin score of w and will not increase $N(a, a'; \{v_\ell^*\})$, even if $b = a'$.

Finally, consider alternatives b where $Inc_\ell(b, a)$ and $Inc_\ell(b, w)$ (note that for all other cases, b must be placed above a, between a and w, or below w). If $b \neq a'$, then we place b below w—this will not change $N(a, a'; \{v_\ell^*\})$ and does not decrease the maximin score

of w. If $b = a'$ and a' is not above a in v_ℓ^*, then we move it above a, as this decreases $N(a, a'; \{v_\ell^*\})$ by 1 and decreases maximin score of w by at most 1, hence their difference can only increase. In all cases above, we can reconfigure votes v_ℓ^* into $\mathbf{v}'$ such that $N(a, a'; \mathbf{v}^*) \geq N(a, a'; \mathbf{v}')$ and $s_m(w, \mathbf{v}^*) \leq s_m(w, \mathbf{v}')$, hence $s_m(w, \mathbf{v}') - N(a, a'; \mathbf{v}') \geq s_m(w, \mathbf{v}^*) - N(a, a'; \mathbf{v}^*) = M_{a'}$. Of course, the construction $\mathbf{v}'$ does not require using $\mathbf{v}^*$ as a starting point—this only serves to prove the optimality of $\mathbf{v}'$ as constructed by the algorithm above.

By this method, we see that the relevant completion for a' given $\mathbf{p}$ can computed in $O(nm)$ time. By Eq. 3.11, pairwise max regret can be computed in time $O(nm^2)$, by computing $M_{a'}$ for all $a' \neq a$ and choosing the largest such value. Consequently, computing minimax regret using this algorithm takes $O(nm^4)$ time:

Theorem 5. *Minimax regret can be computed in $O(nm^4)$ time for the maximin voting rule.*

3.2.4 Bucklin Voting

As with maximin, the Bucklin score of a candidate is not decomposable across voters, so it too requires a more intricate approach to constructing profile completions. Since the standard definition of the Bucklin score involves selection of the alternative with the smallest Bucklin score, we work with the inverted score $b(a, \mathbf{v}) = m - s_B(a, \mathbf{v})$ to remain consistent with our "score maximizing" definitions.

As above, we wish to compute $PMR(a, w, \mathbf{p})$ with respect to (inverted) Bucklin score $b(\cdot, \mathbf{v})$. To compute PMR, we solve the following decision problem, for various values of t: is $PMR(a, w, \mathbf{p}) > t$. The ability to solve this problem efficiently means we can compute PMR in polynomial time by solving this problem for $t = m-2, m-3, \ldots, 0$, in this order. $PMR(a, w, \mathbf{p})$ is then the largest t^* for which $PMR(a, w, \mathbf{p}) > t^* - 1$. (Note that for Bucklin, PMR must be one of $\{0, \ldots, m-1\}$.)

This decision problem can be further broken down by solving a sequence of subproblems which asks whether there is a completion $\mathbf{v} \in \mathsf{C}(\mathbf{p})$ such that $s_B(a, \mathbf{v}) > k$ and $s_B(w, \mathbf{v}) \leq k'$ for fixed values $k, k' \in \{0, \ldots, m-1\}$. If such a completion exists, this implies $PMR(a, w, \mathbf{p}) > k - k'$. Hence, to solve the PMR decision problem above for any fixed t, we can solve the subproblem for all values $k \leq m-1$, with its corresponding $k' = t - k$.

To solve the sub-problem, we consider the relevant worst-case completions of each partial vote, based on different cases, as we did for positional scoring and maximin voting. If either $Nec_\ell(w \succ a)$ or $Inc_\ell(a, w)$ holds, then the worst-case completion places

w as high as possible and a as low as possible in ℓ's vote, as in the case of positional scoring rules.

If $Nec_\ell(a \succ w)$ holds, then more subtle reasoning is required. First, it is clear that all alternatives in the set D (see Figure 3.2, $a \succ w$) must be placed below w, and those in set C above a. This, of course, only serves to improve w's Bucklin score and possibly worsen a's score, which is our aim. At issue, however, is how many alternatives from the set U should be placed above a and how many below w, since we have a choice (obviously none should be placed between a and w). Consider the following two conditions:

- Condition 1: a is ranked below the kth position.
- Condition 2: w is ranked at or above the k'th position.

We consider five distinct cases (in all cases, we assume that all alternatives apart from those in U are now positioned in the appropriate positions):

1. If p_ℓ cannot be arranged such that either Condition 1 or 2 is satisfied (i.e., there are not enough alternatives in U to push a below position k or to push w to or above position k'), then we complete p_ℓ in arbitrary fashion.
2. If p_ℓ can be arranged to simultaneously satisfy both conditions, we use that completion.
3. If p_ℓ can be arranged to satisfy Condition 1 but not Condition 2, we use that completion.
4. If p_ℓ can be arranged to satisfy Condition 2 but not Condition 1, we use that completion.
5. Otherwise p_ℓ can be arranged to satisfy either of Conditions 1 or 2, but not both simultaneously. We tentatively "skip" these partial votes and complete them as the last step of the algorithm.

To solve the subproblem, we use two counters: c_a keeps track of the number of partial votes that fall into Cases 1–4 above in which a's rank is below k; and c_w counts the number of partial votes that fall into Cases 1–4 above in which w's rank is at or above k'. Notice that partial votes within these first four cases offer no scope to trade off the relative position of a and w to satisfy our two conditions. Once these are counted, we need to construct suitable completions of the remaining partial votes, all of which fall into Case 5. Specifically, to validate the subproblem, we need $\lfloor n/2 \rfloor + 1 - c_a$ of the Case

5 votes to be completed in such a way that a's rank is below k, and $\lfloor n/2 \rfloor + 1 - c_w$ votes to be completed so that w's rank is at or above k'. Note that these two requirements are exclusive: the completion of a Case 5 partial vote 5 can only satisfy one of these two requirements.

Now if the number of partial votes n_s satisfying Case 5 is such that $n_s < 2\lfloor n/2 \rfloor + 2 - c_a - c_w$, then we cannot construct the required number of partial votes to satisfy the Bucklin score requirements, therefore we conclude that answer to the sub-problem is false. Otherwise we can satisfy the requirements, by configuring $\lfloor n/2 \rfloor + 1 - c_a$ of these n_s votes so that a's position falls below k (by placing alternatives from U above a), and configuring $\lfloor n/2 \rfloor + 1 - c_w$ of the votes so that w's position is at least in rank k' or better (by placing U below w).

Testing which of the five cases a particular p_ℓ belongs to can be done in $O(m)$ time (see Figure 3.3): we first check if U is large enough to place sufficiently many alternatives above a to make its rank position greater than k. If this is not possible, we then check whether positioning U below w pushes w to position k' or above. If this can be achieved the partial vote falls in case 4, if not it falls in case 1. If a can be pushed below position k, then we check whether the remaining alternatives in U (i.e., those left after removing just enough alternatives to push down a's position) can be placed below w to make its position at least k'. If so, then it falls into case 2. If not, we ask whether the original set U can be placed below w to meet its positional requirement. If so, it falls into case 5, and otherwise it is case 3. Therefore the running time to solve the above sub-problem is $O(nm)$. Since we need to solve these sub-problems at most $m - 1$ times, for value of t of the PMR decision problem, the total time to computing PMR is $O(nm^3)$. Consequently, given m^2 pairs of alternatives, we have:

Theorem 6. *Minimax regret can be computed in $O(nm^5)$ time for the Bucklin voting rule.*

3.2.5 Egalitarian Voting

We've seen rather straightforward computation for PMR and minimax regret using the independent completion of the partial votes of each voter for decomposable rules (positional scoring), while non-decomposable rules like maximin and Bucklin require the "coordinated completion" of different voters' partial votes. However, certain non-decomposable scoring functions do admit the independent completion of partial votes. Consider the egalitarian (or maxmin fairness) voting rule, where $s_f(a, \mathbf{v}) = \min_{\ell \in N}\{m - v_\ell(a)\}$. While minimizing or maximizing the score of a candidate in partial vote p_ℓ is straightforward,

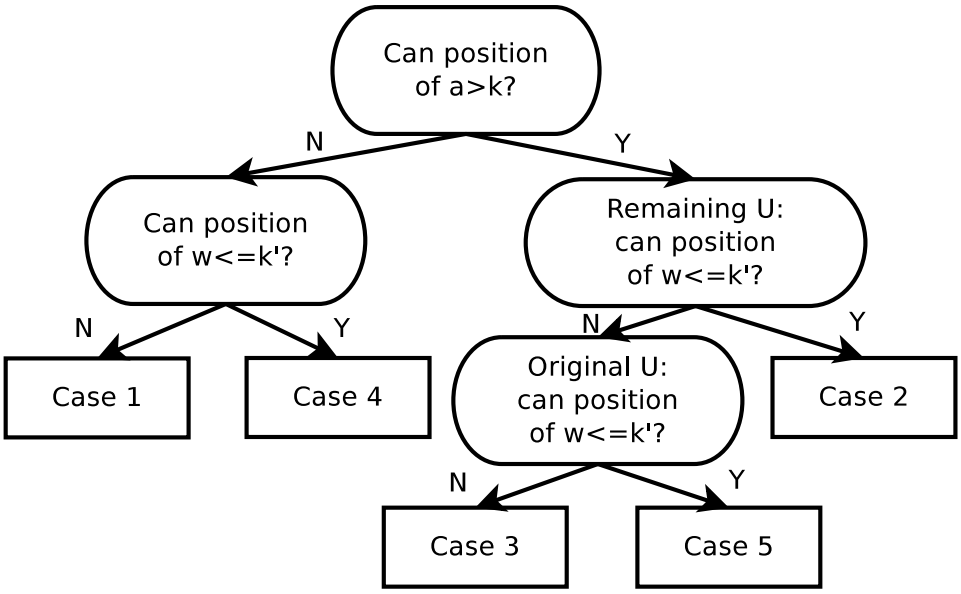

Figure 3.3: Classifying p_ℓ, with $Nec_\ell(a \succ w)$, into one of the five cases, used in answering the sub-problem of whether there exists a completion $\mathbf{v}$ such that $s_B(a, \mathbf{v}) > k$ and $s_B(w, \mathbf{v}) \leq k'$. The decision tree asks whether alternatives in U can be arranged to be placed above a or below w so as to achieve the desired rank positions. See Figure 3.2 for the definition of U.

the way in which *adversarial advantage* is maximized in p_ℓ can depend on other votes. In the cases $Nec_\ell(w \succ a)$ and $Inc_\ell(a, w)$, there is only one way to maximize the local advantage of w over a (see above). But when $Nec_\ell(a \succ w)$, the placement of U either above a or below w influences the maxmin fairness score of a and w in a way that depends on other votes. However, one can show that unless $PMR(a, w, \mathbf{p})$ is negative, then advantage is maximized by ordering U *below* w. Informally, placing U below w can improve the minimum score of both a and w. However, this placement can only improve the minimum score of a if vote v_ℓ gives a its minimum score over all p_ℓ, in which case the minimum score of w is strictly less than that of a, and $PMR(a, w, \mathbf{p})$ is negative. Since max regret can never be negative, the pair (a, w) cannot define a's max regret. This lets us prove that, unless PMR is negative, $PMR(a, w, \mathbf{p})$ is maximized by ordering U below w in any p_ℓ where $Nec_\ell(a \succ w)$. In other words, we maximize the positional score of w without concern for the score of a. The running time for PMR is $O(nm)$ and for MMR is $O(nm^3)$, as in the case of positional scoring rules, since we only need to identify the relevant sets in Figure 3.2.

Theorem 7. *Minimax regret can be computed in $O(nm^3)$ time for egalitarian voting.*

3.3 Preference Elicitation

The ability to compute regret-minimizing winners is critical when only partial information is available about voter preferences. However, in many cases, one has the option of eliciting additional preference information from the voting population in order to reduce minimax regret and improve decision quality. This can be especially important when the minimax regret of the partial profile exceeds some (problem- or domain-specific) tolerance, indicating the potential for divergence from optimality beyond an acceptable threshold.

Of course, as discussed above, elicitation can be difficult (with respect to both computation and communication complexity) in the worst case. Despite this, we will show how minimax regret can be used to effectively guide the elicitation process, providing strong results in practice. As a measure of solution quality, minimax regret can be used to terminate elicitation whenever regret falls to some suitable threshold (including zero if optimality is desired).[5] More importantly, the solution to the minimax optimization can guide the selection of queries (and voters) so that an (approximate or exact) optimal solution can often be found quickly. In this section, we describe a simple heuristic strategy to do just this. We focus on linear positional (Borda-like) rules and two specific query types; but these ideas generalize to other rules and other forms of queries.

The Current Solution Strategy (CSS)

We consider a general technique, called the *current solution strategy (CSS)*, that has been used effectively in both single-agent recommendation systems [14, 118, 19], multi-agent mechanism design [65, 67, 66], and in multi-agent stable matching [44] settings (the latter using the completion techniques we develop in this paper). Intuitively, CSS identifies voter preference information that helps assess the relative degree of preference between the minimax optimal solution and the adversarial witness within each voter's partial preference ranking, and queries a voter whose response has the greatest potential to reduce the advantage of the witness over the minimax optimal alternative.

The precise instantiation of CSS depends on the types of preference queries permitted. We consider two forms of queries in this work. A *comparison query* identifies a voter ℓ, and asks ℓ to compare two alternatives: "Is $a \succ_\ell b$?". A *top-k query* identifies and asks voter ℓ to state which alternative is kth in their ranking (we assume that the first $k-1$ alternatives have already been articulated by ℓ). We describe CSS in detail using

[5] If determining optimal termination is hard for a particular voting rule [32, 121], then so is computing minimax regret; or equivalently, if computing minimax regret is easy (as demonstrated for certain rules above), so is termination.

comparison queries, but we show how the same intuitions can be adapted to the selection of top-k queries.

Assume we are allowed to ask any voter a pairwise comparison query. CSS generates queries by considering the current solution to the minimax optimization—i.e., the minimax optimal alternative a and adversarial witness w—and using this to choose a voter-query pair with greatest potential to reduce minimax regret. Notice that if the advantage of w over a is not reduced in some partial vote p_ℓ in response to a query, $PMR(a, w)$ will not change, thus, unless the response changes the minimax optimal solution, MMR will not change. So CSS selects queries that tackle this gap directly. We determine the value of posing a query to voter ℓ by considering the three cases in Figure 3.2, in each case determining the query with the largest potential reduction given a positive response by ℓ:

Case 1, where $a \succ w$: Recall that a worst-case completion must place (all alternatives in) set D below w, set C above a, and set U either below w or above a. We can reduce $PMR(a, w)$ by asking two different types of queries: $d \succ w$ for some $d \in D$ or $a \succ c$ for $c \in C$. In each case, a positive response will position alternatives between a and w, thus reducing $PMR(a, w)$ by increasing the (worst-case) position of a relative to w in p_k, Specifically, a positive response to $d \succ w$ prevents the adversary from placing d, or any of its ancestors in D, below w in the completion. And a positive response to $a \succ c$ prevents the adversary from placing c, or any of its descendents in D, above a in the completion. We pick the alternative in $C \cup D$ (and corresponding query) with greatest potential to reduce PMR in the case of such a positive response. For linear scoring rules, this potential is measured by the number of ancestors of d in set D (all of which will be positioned between a and w if d is), and the number of descendents of c in set C. If $C \cup D = \emptyset$, we can ask two other query types, $u \succ w$ or $a \succ u$ for some $u \in U$. These do not reduce PMR directly, but shift u and its ancestors in U to set C (for query $u \succ w$) or u and its descendents in U to set D (for query $a \succ u$). The u with the potential to shift greatest number of alternatives within U to some other set is chosen.

Case 2, where $w \succ a$: Recall that a worst-case completion must place sets E, A and U between w and a. We can reduce $PMR(a, w)$ by asking four different types of queries: $a \succ e$ for some $e \in E$; $a \succ u$ for $u \in U$; $x \succ w$ for some $x \in A$; or $u \succ w$ for some $u \in U$. A positive response to any such query will reduce PMR. In the case of query $a \succ e$ (resp., $a \succ u$), it increases the (worst-case) score of a in p_ℓ, by preventing the adversary from positioning e (resp., u), or any of its descendents in E (resp., U), between w and a. Similarly, in the case of query $x \succ w$ (resp., $u \succ w$), it reduces the (worst-case) score of

w in p_ℓ, by preventing the adversary from positioning x (resp., u), or any of its ancestors in A (resp., U) between w and a. Selection is again made by choosing the alternative and query that have the greatest potential (i.e., the number of alternatives that are removed from their respective sets E, A or U in case of a positive response).

Case 3, where $Inc(a, w)$: We can reduce $PMR(a, w)$ by asking several different queries, however, heuristically, we always choose to ask if $a \succ w$, since a positive response reverses p_ℓ's contribution to PMR from positive to negative. *Any* response will move partial vote p_ℓ into either case 1 or case 2.

We note that each voter is "scored" by assessing which query has the greatest potential to reduce adversarial advantage in her partial vote, based on which of the three cases above by which her partial is classified. When asking a single query at each round, we choose the voter whose corresponding best query has the greatest potential (breaking ties arbitrarily).

CSS must eventually terminate with an optimal solution:

Proposition 1. *Unless* $MMR(\mathbf{p}) = 0$*, CSS will always select a voter* ℓ *and comparison query* $a_i \succ_\ell a_j$ *such that* $Inc_\ell(a_i, a_j)$*.*

Proof. Suppose that $MMR(\mathbf{p}) > 0$, that a is a minimax optimal alternatives, and w is its adversarial witness. If there exists a partial vote p_ℓ such that $Inc_\ell(a, w)$ then CSS will be able to query that voter with $a \succ w$. If there is a partial vote p_ℓ such that $Nec_\ell(a \succ w)$ and at least one of the sets D, C or U is non-empty, then CSS will generate a query (see Case 1 in the CSS description). If there is a partial vote p_ℓ such that $Nec_\ell(a \prec w)$ and at least one of the sets F, E or U is non-empty, then again CSS will generate a query (see Case 2).

Finally, suppose no partial vote satisfies the three conditions above. This implies that all partial votes fall into Case 1 or Case 2, and all Case 1 votes satisfy $D \cup C \cup U = \emptyset$ and all Cases 2 votes satisfy $F \cup E \cup U = \emptyset$. However, this means the exact positional scores of a and w can be determined in every partial vote. By definition of MMR, w must have a strictly higher score than a in some completion if $MMR(\mathbf{p}) > 0$. But this means w has a strictly higher score than a in *every* completion of $\mathbf{p}$, contradicting the minimax optimality of a. □

CSS can be adapted, using similar intuitions, to generate top-k queries. Such queries are asked of each voter in order—no no voter is asked for the second-ranked candidate before their revealing their first-ranked candidate, their third before their second, etc. This means that CSS need only select a voter at any stage—the query is determined

once the voter is selected. If elicitation is confined to top-k queries, for a given voter we consider four cases to assess the potential reduction in MMR associated with querying that voter (again assuming minimax optimal a and adversarial witness w):

Case 1: If a and w are already in the voter's top-k list, then their scores are known with certainty and $PMR(a, w)$ cannot be reduced by querying this voter.

Case 2: If a is in the voter's top-k list but w is not, asking for the next highest alternative—assuming optimistically that the response is not w—reduces the adversarial advantage by one (i.e., the adversary would place w at rank $k + 2$ instead of $k + 1$).

Case 3: If a is not in top-k list, but w is, the adversary maximizes PMR by placing a at the bottom of the ranking. If a is (optimistically) returned as the next highest alternative, PMR is reduced by $m - (k + 1)$ (since a must be at position $k + 1$).

Case 4: If neither a nor w are in the top-k list, the adversary will place a at the bottom of the ranking and w at position $k + 1$. If a is (optimistically) returned as the next highest alternative, PMR is reduced by $m - k$ (a must now be placed at position $k + 1$ and w at $k + 2$).

In each of the cases above, we calculate the potential PMR reduction for each voter, and query the voter with the largest potential reduction. It is clear that, unless minimax regret is zero, CSS will always have some voter to query with positive potential reduction.

Proposition 2. *Unless $MMR(\mathbf{p}) = 0$, CSS for top-k queries will always select a voter ℓ to query for its next most preferred alternative.*

Proof. We argue that, if $MMR(\mathbf{p}) > 0$, there is at least one agent ℓ that does not belong to Case 1 above (i.e., where minimax winner a and its witness w both appear in an agent's current list of most preferred alternatives). Suppose this is false. Then *all* agents have specified the rank positions of both a and w, hence the positional scores of a are w are both fully known. We argue that since MMR is strictly positive, w must have lower max regret than a. Since MMR is positive, and we know the rank position of w for all agents, then the positional score of w must be greater than a. That is, there exists a positive constant $C > 0$ such that

$$s(w, \mathbf{v}) - s(a, \mathbf{v}) = C, \quad \forall \mathbf{v} \in \mathsf{C}(\mathbf{p}). \tag{3.12}$$

We also know that for any other alternative $b \neq a$,

$$\begin{aligned} PMR(a, b, \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(b, \mathbf{v}) - s(a, \mathbf{v}) \\ &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(b, \mathbf{v}) - s(w, \mathbf{v}) + s(w, \mathbf{v}) - s(a, \mathbf{v}) \\ &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(b, \mathbf{v}) - s(w, \mathbf{v}) + C \quad \text{by Eq. 3.12} \\ &= PMR(w, b, \mathbf{p}) + C. \end{aligned}$$

Hence $PMR(w, b', \mathbf{p}) < PMR(a, b', \mathbf{p})$ for all $b \neq w$. This implies that $MR(w, \mathbf{p}) < MR(a, \mathbf{p})$ which would imply that a is *not* the minimax winner. This is contradicts our assumption and therefore there is at least one agent ℓ that belongs to Case 2, 3, or 4, and hence CSS would query for the next most preferred alternative for ℓ. □

3.4 Empirical Evaluation

We now evaluate the performance of CSS, for both pairwise comparisons and top-k queries, on a few datasets. In particular, we are interested in the amount of preference information that CSS elicits before the minimax regret is small enough. We compare CSS against two other baseline elicitation strategies. The *random strategy (Rand)* randomly chooses a voter ℓ and a comparison query such that $Inc_\ell(a_i, a_j)$ (so the query response always bears information). With top-t queries, Rand only needs to choose voter ℓ at random. The *volumetric strategy (Vol)* selects a voter ℓ and query $a_i \succ a_j$ that maximizes the number of *new* pairwise preferences revealed (given the worst response):

$$Vol(p_\ell) = \max_{a_i, a_j} \min \left\{ \begin{array}{l} |\mathsf{tc}(p_\ell \cup \{a_i \succ a_j\})|, \\ |\mathsf{tc}(p_\ell \cup \{a_j \succ a_i\})| \end{array} \right\},$$

where tc denotes transitive closure. This strategy reduces preference uncertainty maximally, without regard for the "relevance" of the revealed preference information to winner determination (much like volumetric strategies for in polyhedral conjoint analysis for single-agent problems [115]). Its application to top-k queries involves selecting the voter whose next-ranked candidate reduces uncertainty the most. Since this voter *must* be one who has ranked the fewest candidates, the strategy reduces to a simple sequential iteration: each voter in turn is asked for their top-ranked candidate; then each is asked for their second-ranked candidate; and so on. We refer to Vol in this case as SequentialTop.

In related work (see Section 3.5), Kalech et al. [69] proposed two vote elicitation

algorithms. Their first method is essentially the SequentialTop method described above, and proceeds in rounds in which each voter is queried for their next most preferred choice. It uses necessary winner computation for termination. This contrasts with our CSS approach, which is much more subtle and incremental: we identify a *particular voter* to query at each stage by evaluating its potential to reduce minimax. We see in our experiments that this can reduce the number of required queries substantially. Furthermore, our elicitation methods are anytime: querying can terminate when minimax regret is sufficiently small, and we show below that this further reduces the number of queries significantly.

Kalech et al.'s [69] second algorithm proceeds for a predetermined number of rounds, asking each voter at each stage for fixed number of positional rankings. Since termination is predetermined, necessary winners may not result (instead possible winners are returned), and interesting tradeoffs between the number of rounds and amount of information per round are explored. One attractive feature of this model is the batching of queries (voters are only queried a fixed, ideally small, number of times, though each query may request a lot of information), thus minimizing interruption, waiting time, etc. As the authors acknowledge, this scheme provides no guarantee of winner optimality or any bounds on quality. A key advantage of our minimax regret-based scheme is that a natural, precise objective is being minimized, and anytime quality guarantees are provided.[6] This second approach of Kalech et al. does, however, batch queries, so voters are only queried a few times, though each query may request more information than the CSS scheme. Such batching can minimize user interruption as well as user *latency* (since voters are not required to wait until the responses of other voters are delivered before their next query is received). We will return to the issue of batching in Chapter 6.

We test CSS on three different data sets:

- *Sushi*: The Sushi data set [70] contains 5000 full preference rankings over 10 varieties of sushi.

- *Irish*: The Irish data set comprises the votes of the 2002 Irish national election from the Dublin North constituency. It contains $43,942$ ballots of the top-t form over 12 candidates. We use the subset consisting of the 3662 complete ballots (i.e., that rank all 12 candidates).[7]

[6] We also note that elicitation of pairwise preferences is not considered in [69]; such queries are extremely valuable and arise naturally in many domains such as search, information retrieval, consumer product comparisons, etc.

[7] See `www.dublincountyreturningofficer.com`.

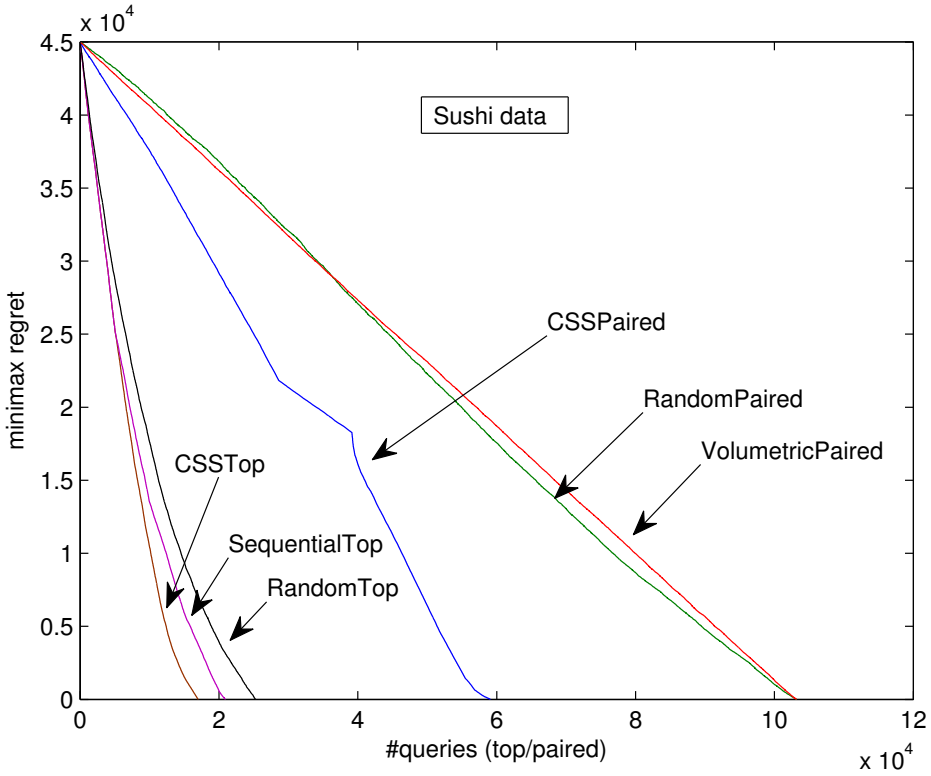

Figure 3.4: Performance of elicitation algorithms (paired and top-k queries) on Sushi.

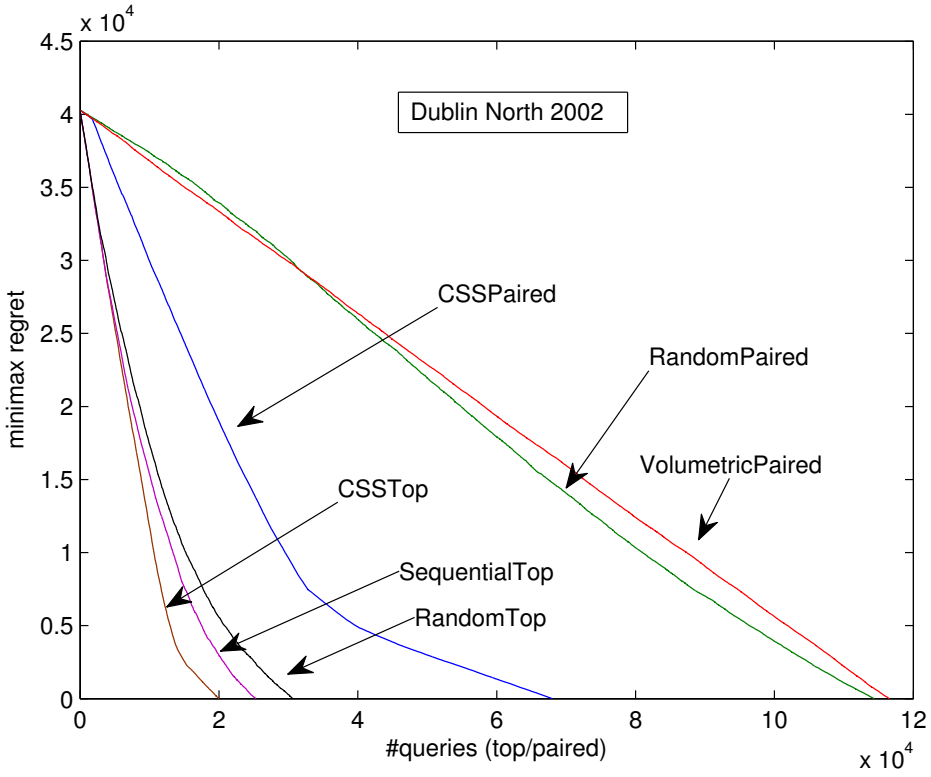

Figure 3.5: Performance of elicitation algorithms (paired and top-k queries) on Irish.

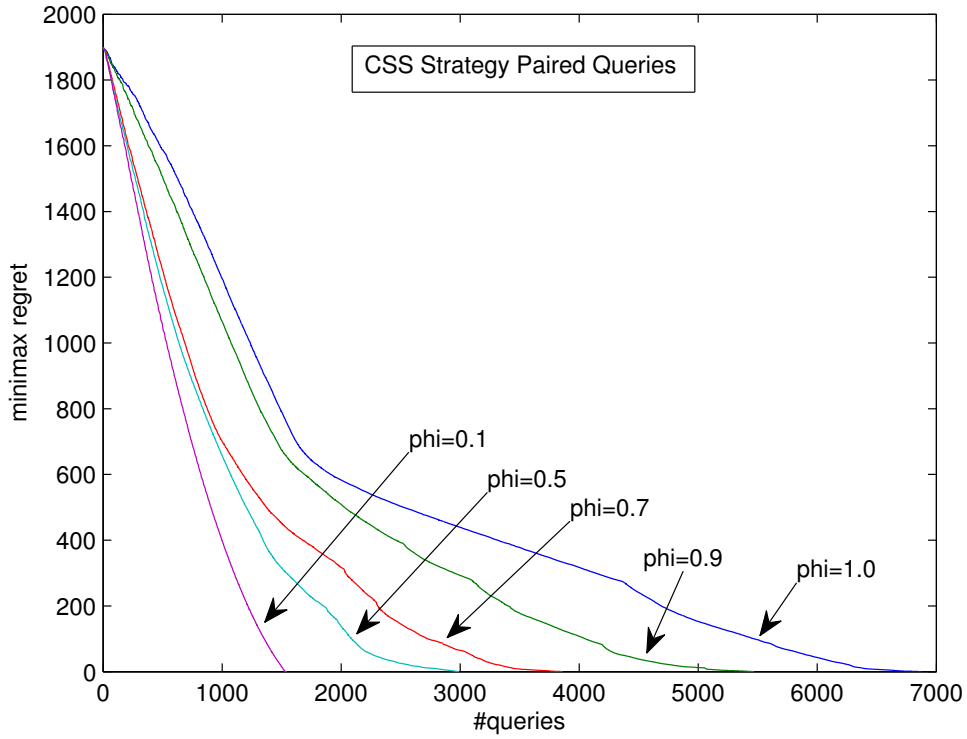

Figure 3.6: Performance of CSS elicitation (paired) on Mallows data.

- *Mallows*: We generate random vote profiles, each with 100 random rankings over 20 alternatives, where each ranking is drawn from a Mallows ϕ-model, using several different parameter settings (as we explain further below). Note that in Mallows, a smaller ϕ concentrates probability mass around the modal ranking σ (i.e., all voters have nearly identical rankings), while $\phi = 1$ gives the uniform distribution over rankings, also known as *impartial culture*.

These data sets were used to generate responses to elicitation queries, and reflect different forms of group decisions, both political voting and the types of group decisions often needed in recommender systems.

We test CSS on each data set, measuring how quickly minimax regret reduces as a function of the total number of queries asked of the voting population. For the real-world data (Sushi, Irish), we use both pairwise comparisons and top-k queries, and compare the performance of CSS to both the random and volumetric elicitation strategies. On the Mallows (random) data, we use only pairwise comparisons and analyze the impact on convergence as we vary the dispersion parameter. In all cases, we use Borda scoring to measure candidate quality (similar results hold for other rules).

The plots in Figures 3.4 and 3.5 shows *MMR* as a function of the total number of

queries asked on the Sushi and Irish data sets. We see that CSS offers superior elicitation performance with both pairwise comparisons and top-k queries. With Sushi, CSS reaches the optimal solution (i.e., the provable winner with $MMR = 0$) after an average of only 11.82 comparison queries per voter (or a total of 59,100 queries). This compares with 20.64 queries for Vol and 20.63 queries for Rand (as well as the 25 queries required by the theoretically optimal MergeSort to determine full voter rankings). With top-k queries, CSS needs only 3.40 queries per voter to reach $MMR = 0$, compared to 4.18 for Seq, and 5.50 for Rand. With Irish, results are similar: CSS reaches optimality with 18.57 comparison queries and 5.47 top-k queries per voter. This stands in contrast with 31.82 comparisons and 6.91 top-k queries for Vol/Seq; 31.22 comparisons and 8.38 top-k queries for Rand, and 33 comparisons for MergeSort. Note that top-k queries are "information rich" as they provide many pairwise comparisons per response. Thus, while CSS's advantage is somewhat less in the top-k case (though RandomTop still requires over 50% more queries to reach $MMR = 0$ in Irish), the fact that there is an advantage is of greater significance due to the greater "intensity" of each query.

Critically, if one is interested in approximate solutions, we see that CSS reduces MMR very quickly, providing high-quality solutions after very few queries. For example, with Irish, CSS reduces MMR to 18% of its initial value (with no voter preference data) after only 5.82 comparison queries per voter, which is a small fraction of the queries required to elicit full rankings. By contrast, to reduce regret to the same degree requires 25.77 comparisons for Vol, and 24.03 comparisons for Rand. Computationally, for problems of this size, CSS takes only a few milliseconds on average (wall clock time) to find the best agent/comparison query (including time needed to recompute the MMR-solution).

On the synthetic Mallows data set, we sample 100 complete voter profiles, each with 100 voters and 20 alternatives, for each of several different values of ϕ. The same impact of CSS on MMR as a function of the number of pairwise comparisons asked is shown in Figure 3.6. With larger ϕ, more queries are clearly needed to reach the same level of regret, which conforms to our intuitions that intelligent elicitation schemes can take significant advantage of less uniform preferences to minimize queries and voter effort (and conversely, that with almost uniformly random preferences, nearly full rankings must be obtained). Work in behavioral social choice strongly suggests that real-world preferences are not uniformly random [106], and CSS seems to perform especially well in this case; indeed our results on Sushi and Irish suggest that real preferences are not uniform, and contain regularities that can be readily exploited to reduce the informational complexity of voting.

3.5 Related Work

The *elicitation* question has been studied from a theoretical perspective, addressing whether winners for some voting rules can be determined with partial voter preferences (rankings). Unfortunately, worst-case results are generally discouraging. Conitzer and Sandholm [34] demonstrate that the communication complexity of several common voting protocols, such as Borda and Copeland, is $\Omega(nm \log m)$, essentially requiring communication of full voter preferences in the worst-case. Indeed, determining which votes to elicit to determine a winner is NP-hard in many schemes (e.g., Borda) [32, 121].

Despite the theoretical complexity of partial elicitation, practical means of eliciting partial rankings and making decisions with partial preferences are vital. This is precisely the problem we address in this chapter. Kalech et al. [69] were among the first to consider practical vote elicitation schemes for a couple of prominent score-based voting rules (i.e., Borda and range voting). They propose two elicitation schemes using specific query types, and adopt possible and necessary winners as their primary solution concept. Their *iterative voting* method—this is distinct from iterative voting schemes in which voters change their votes in response to the current vote profile, which go by the same name—determines a true winner (with no approximation) and proceeds in *rounds*. At each round, voters are queried for their next best candidate (and by round k have answered the equivalent of a top-k query, as defined below). At the end of each round, necessary and possible winners are computed with respect to the current partial vote profile: if every possible winner is a necessary (co-)winner, the process stops and returns the set of necessary winners. (The algorithm could also be terminated once *any* necessary winner is found.) On small random and real-world vote profiles (up to 30 voters and 50 alternatives), this scheme can reduce the number of alternatives ranked by voters by up to 10–40%, with the larger gains possible when user preferences are more uniform.

Kalech et al. [69] also propose *greedy voting*, which proceeds for a fixed number of rounds k. Given the current partial profile $\mathbf{p}^k$ at round k, the minimal and maximal possible scores $s(a, r)$, over all completions r of $\mathbf{p}^t$, of each candidate a are computed. Then each voter ranks the set of q alternatives with the largest minimum scores (maximum scores are used to break ties), for some small q. Since termination must occur after k rounds, necessary winners may not result, so possible winners are returned. This model batches queries (voters are only queried a fixed, ideally small, number of times, though each query may request more information than in the other schemes discussed below). Such batching can minimize user interruption as well as user *latency* (since voters are required to wait until the responses of other voters are delivered before their next query

is received). This scheme cannot guarantee winner optimality, nor any bounds on quality. Kalech et al. show empirically that the (post hoc) quality of the resulting winners in small domains is reasonably high, and of course improves with the number of rounds permitted.

Subsequent work on elicitation has focused on several rather distinct approaches. Soufiani et al. [114] exploit a probabilistic model of voter utilities given attributes of both voters and candidates, and applying a Bayesian experiment design approach to elicit full rankings. However, their scheme is not directly driven by a specific decision criterion. Ding and Lin [42] analyze the computation of a "deciding set" of queries for a particular candidate—these are queries where any consistent set of responses will fully determine whether a candidate must be a winner or loser. However, this cannot be construed as an interactive elicitation scheme.

Finally, we note that *minimax regret*—the robustness criterion we adapt to the problem of winner determination under partial profiles and to drive our elicitation process—has been used rather widely for the same purposes in single-agent decision making and preference/utility elicitation. This includes work on multi-attribute optimization problems [12, 109, 122, 13, 14, 18, 15, 19], combinatorial optimization of item slates [118], and Markov decision processes [104, 105]. It has also been used in (multi-agent) mechanism design to minimize the amount of utility function information revealed to direct mechanisms such as the VCG mechanism [65, 67, 66]. We refer to [11] for an overview. We also note that minimax regret has been used as a robustness criterion for optimization problems (e.g., linear programs) in which there is data or objective function uncertainty [77, 5, 3].

3.6 Conclusion

In this chapter we have introduced and defined the concept of *minimax regret (MMR)* as the decision criterion for making robust social choice decisions with an arbitrary collection of partial preferences. The MMR criterion and the notion of max regret allows a decision maker to assess the difference in the quality of any outcome with respect to the (unknown) optimal outcome. Specifically, it lets us bound the loss in "social welfare" associated with the underlying voting rule. Consequently, this approach supports the informational approximation of voting rules. We compared MMR to possible and necessary winners, showing that it not only generalizes necessary winners but that it also provides a worst-case guarantee on the decision quality. We also showed through examples that the notion of possible winners is not supportive of decision making, and that while an alternative is

not a possible winner it may still have good max regret.

We offered basic observations about the time complexity of computing MMR optimal alternatives. We gave practically efficient, polynomial time algorithms for computing MMR for some common voting rules including all positional scoring rules (such as Borda, k-approval, etc.), Maximin, Bucklin and Egalitarian.

In preference elicitation, MMR also serves as the driving objective of any querying strategy. In particular, we developed an elicitation strategy known as the current solution strategy (CSS) for pairwise and top-k queries that quickly reduces MMR. Experiments on real and synthetic datasets showed that CSS is superior to a "volumetric" strategy and random querying for both pairwise and top-k queries. Experiments also showed that in practice, only a fraction of preference information is required to make optimal or near-optimal decisions.

While this chapter has covered the elicitation of preferences over *atomic* alternatives, that is, alternatives that have no inherent structure, in many real application domains the alternatives are multi-attributed and large in number. For example, this is the case in recommender systems, such as recommending apartments or cars (which may consist of thousands of alternatives). Preferences over such alternatives are usually more structured, and exploit the fact that, for example, each agent's utility can be approximated by the sum of the utility of each individual attribute. Because there is much more structure in multi-attributed domains, the robust optimization computations are more sophisticated and preference elicitation queries would involve attributes of alternatives. However, with added structure, we may be able to develop faster robust optimization algorithms and improved elicitation schemes.

Another interesting avenue for future work is to explore different query types for preference elicitation. In this chapter we have focused on pairwise comparisons and top-k queries. However, there are a few other query types that appear natural and perhaps more sensible in specific applications for certain individuals. This includes querying for both top- and bottom-k (many people can often easily identify their most and least preferred alternatives), querying for the most preferred alternative from a smaller subset of alternatives, or perhaps even asking for alternatives at particular rank positions. Such queries, or mixing of these query types would require the development of new elicitation strategies.

Chapter 4

Robust Optimization and Elicitation for Multiple-Choice Problems

Having introduced the regret-based decision criterion for single-choice problems, we now turn our attention to the multi-winner problems, and consider the robust optimization of a *slate* of alternatives given a partial preference profile, again using minimax regret as our robustness criterion.

We begin in Section 4.1 by defining and discussing the multi-winner voting rule that we study in this Chapter. Section 4.2 introduces and defines minimax regret for multi-winner problems. We discuss the computation of minimax optimal slates in Section 4.2.1, focusing our attention on linear positional scoring rules (such as Borda) for ease of exposition. We describe the relevant completion principles, and show that the computational problem is hard in general, but can be solved in polynomial time for bounded slate sizes K (i.e., it is fixed-parameter tractable with respect to K). In Section 4.2.2 we describe a greedy algorithm for approximating minimax-optimal slates and prove approximation ratios. We then explore, in Section 4.3, the use of these methods to drive preference elicitation strategies for multi-winner problems. We test the empirical performance of our greedy slate selection method and of our preference elicitation strategy on some real and synthetic datasets in Section 4.4. Finally, we conclude in Section 4.5.

4.1 Preliminaries

Multi-winner problems deal with the selection of a set or *slate of alternatives* $\overline{a} \subseteq A$. A voter's satisfaction with a slate is a function of their satisfaction with the alternatives in the selected slate, and constraints are generally placed on the set of *feasible slates*. In this chapter, we focus on the conceptually simplest such multi-winner model: (i) we

assume that voter satisfaction with candidates is given by some scoring function (e.g., the Borda PSF) induced by their ranking, and that a voter's satisfaction with the slate is given by the score of their most preferred alternative on the slate; and (ii) feasible slates are those with size $|\overline{a}| \leq K$. In other words, we can select up to K candidates and each voter derives benefit from their most preferred.

More formally, given a preference profile $\mathbf{v}$, we define the *score* of a K-set and the *optimal K-set* as follows:

$$s(\overline{a}, \mathbf{v}) = \sum_{\ell \in N} \max_{a \in \overline{a}} \alpha_\ell(a) = \sum_{\ell \in N} s_\ell(\overline{a}), \tag{4.1}$$

$$\overline{a}^*_{\mathbf{v}} = \operatorname*{argmax}_{|\overline{a}| \leq K} s(\overline{a}, \mathbf{v}). \tag{4.2}$$

(We suppress dependence of s on α since the PSF will be fixed and clear from context.) When α is the Borda PSF, this corresponds to the *proportional representation (PR)* scheme of Chamberlin and Courant [25] as described in Section 2.3.[1] For the remainder of this chapter we focus on robust optimization and preference elicitation for multi-winner problems that correspond to the above proportional representation scheme as defined in Section 4.1. An important variant proposed by Monroe [92] requires the use of an assignment function—that associates each voter with a specific winner—so that candidates each represent a roughly equal number of voters.

This slate optimization problem can be viewed as a *segmentation problem* [74]; and it is also a special case of *budgeted social choice*, specifically, the *limited choice* form of the problem [83]. See Section 2.3 for a more detailed discussion. More general forms of proportional representation [92] and budgeted social choice [83] allow for *assignment functions* that map voters to specific alternatives (e.g., to ensure balanced representation, or budget feasibility); but here we assume that the only constraint is on the number of alternatives selected. Sets of size less than K offer no advantage over those of size K in this case.

4.2 Minimax Regret for Slate Optimization

We now consider the problem of selecting an optimal K-set of alternatives when we have only a partial preference profile $\mathbf{p}$ rather than a complete profile $\mathbf{v}$. We again adopt *minimax regret* as a robustness criterion for making decisions with a partial profile.

[1] Note that Chamberlin and Courant [25] specify their model in terms of *misrepresentation* and attempt to minimize total degree of misrepresentation rather than maximize total satisfaction; but the notions are equivalent.

The definitions of pairwise maximum regret, maximum regret, and minimax regret are analogous to those in the single-winner case, with outcomes/decisions (and adversarial choices) corresponding to slates of K alternatives rather than single choices. We first present definitions (for sets of fixed size K) and then explain the intuitions. Recall that $\overline{a}$ and $\overline{w}$ represent slates of alternatives.

$$\mathit{Regret}(\overline{a}, \mathbf{v}) = \max_{|\overline{w}| \leq K} s(\overline{w}, \mathbf{v}) - s(\overline{a}, \mathbf{v}) \tag{4.3}$$

$$\mathit{PMR}(\overline{a}, \overline{w}, \mathbf{p}) = \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} s(\overline{w}, \mathbf{v}) - s(\overline{a}, \mathbf{v}) \tag{4.4}$$

$$\begin{aligned} \mathit{MR}(\overline{a}, \mathbf{p}) &= \max_{\mathbf{v} \in \mathsf{C}(\mathbf{p})} \mathit{Regret}(\overline{a}, \mathbf{v}) \\ &= \max_{|\overline{w}| \leq K} \mathit{PMR}(a, \overline{w}, \mathbf{p}) \end{aligned} \tag{4.5}$$

$$\mathit{MMR}(\mathbf{p}) = \min_{|\overline{a}| \leq K} \mathit{MR}(\overline{a}, \mathbf{p}) \tag{4.6}$$

$$\overline{a}^*_{\mathbf{p}} \in \operatorname*{argmin}_{|\overline{a}| \leq K} \mathit{MR}(\overline{a}, \mathbf{p}) \tag{4.7}$$

Given a vote profile $\mathbf{v}$, $\mathit{Regret}(\overline{a}, \mathbf{v})$ describes the loss in satisfaction associated with offering set $\overline{a}$ rather than the optimal K-set. Given a partial profile $\mathbf{p}$, the *pairwise max regret* $\mathit{PMR}(\overline{a}, \overline{w}, \mathbf{p})$ is the worst-case loss that could be incurred, under all possible realizations of consistent voter preferences, by offering $\overline{a}$ rather than $\overline{w}$. Note that our definition of PMR does not impose constraints on set sizes, a fact we exploit below. The *max regret* $\mathit{MR}(\overline{a}, \mathbf{p})$ of set $\overline{a}$ is the worst-case loss relative to the *optimal* K-set under all preference realizations: this bounds the loss associated with $\overline{a}$ given our preference uncertainty. Finally, a *minimax optimal set* $\overline{a}^*_{\mathbf{p}}$ is one with minimum max regret or *minimax regret* $\mathit{MMR}(\mathbf{p})$.

Observation 3. If $\mathit{MMR}(\mathbf{p}) = 0$, then $\overline{a}^*_{\mathbf{p}}$ is an optimal slate of alternatives for any $\mathbf{v} \in \mathsf{C}(\mathbf{p})$.

4.2.1 Computing MMR-Optimal Slates

Before discussing computation of minimax regret, we begin with the simpler problem of computing pairwise max regret. From Eqs. 4.4–4.6, we see that the regret-optimal slate $\overline{a}^*_{\mathbf{p}}$ can be determined by first computing $\mathit{PMR}(\overline{a}, \overline{w}, \mathbf{p})$ for all pairs of K-sets $\overline{a}, \overline{w}$, maximizing over $\overline{w}$ to determine $\mathit{MR}(\overline{a}, \mathbf{p})$, then minimizing over these terms to compute $\mathit{MMR}(\mathbf{p})$. If K is small, then robust optimization is efficient if PMR can be computed effectively, a problem on which we first focus. (We discuss an approach for large K in

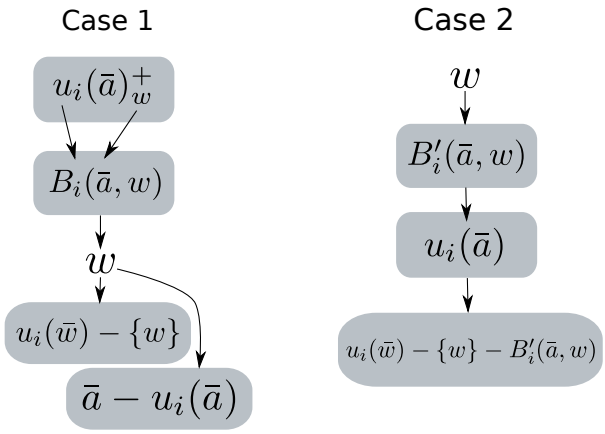

Figure 4.1: Adversarial completions of p_ℓ for $PMR(u_\ell(\overline{a}), \{w\}, p_\ell)$. Case 1, when there exists $a \in \overline{a}$ such that $a \succ_\ell w$, and case 2 when no such a exists. In case 1, PMR is given by Eq. 4.10 and in case 2, PMR is given by Eq. 4.11.

Section 4.2.2.) Just as with single-winner problems using positional scoring rules, one can show PMR is additively decomposable:

$$\begin{aligned} PMR(\overline{a}, \overline{w}, p_\ell) &= \max_{v_\ell \in \mathsf{C}(p_\ell)} s(\overline{w}, v_\ell) - s(\overline{a}, v_\ell) \\ PMR(\overline{a}, \overline{w}, \mathbf{p}) &= \sum_{\ell \in N} PMR(\overline{a}, \overline{w}, p_\ell) \end{aligned} \tag{4.8}$$

Thus we can compute the contributions of each voter ℓ to PMR independently. When ℓ is presented with slate $\overline{a}$, she will choose her most preferred alternative from $\overline{a}$, and similarly for slate $\overline{w}$. Define the *undominated elements* for voter ℓ in any set $\overline{a}$ to be:

$$u_\ell(\overline{a}) = \{a \in \overline{a} : \nexists a' \in \overline{a} \text{ s.t. } a' \succ_\ell a\}.$$

If presented with a slate $\overline{a}$, ℓ's maximal satisfaction—informally, we call this ℓ's *choice* for $\overline{a}$—can only be derived from one of these undominated elements, no matter what completion of ℓ's partial preferences reflects her true underlying preferences. In the ranking $v_\ell \in \mathsf{C}(p_\ell)$ that maximizes pairwise regret, only one element in $\overline{w}$ will be chosen by ℓ (the most preferred), which defines PMR:

$$PMR(\overline{a}, \overline{w}, p_\ell) = \max_{w \in u_\ell(\overline{w})} PMR(u_\ell(\overline{a}), \{w\}, p_\ell). \tag{4.9}$$

Given this, there are two cases to consider when determining the adversarial completion $v_\ell \in \mathsf{C}(p_\ell)$ that maximizes $PMR(u_\ell(\overline{a}), \{w\}, p_\ell)$.

Case 1: Suppose there is an $a \in \overline{a}$ such that $a \succ_\ell w$. This means there is no completion

in which ℓ would choose w, so PMR is negative. Maximizing pairwise regret requires reducing the "gap" between the most preferred $a^* \in \overline{a}$ and w. The only alternatives that must lie between the most preferred a and w are undominated elements $u_\ell(\overline{a})$ of $\overline{a}$ that dominate w, or those b known to lie between such an a and w. Define

$$u_\ell(\overline{a})^+_w = \{a \in u_\ell(\overline{a}) : a \succ_\ell w\},$$
$$B_\ell(\overline{a}, w) = \{b \in A : \exists a \in u_\ell(\overline{a})^+_w, a \succ_\ell b \succ_\ell w\}.$$

$B_\ell(\overline{a}, w)$ includes all alternatives that must lie between the best $a \in \overline{a}$ and w (the specific choice or placement of the elements in these two sets has no impact on PMR). Every other alternative can consistently be ordered above the best a or below w depending on constraints in p_ℓ.[2] Thus we have:

$$PMR(u_\ell(\overline{a}), \{w\}, p_\ell) = -|u_\ell(\overline{a})^+_w| - |B_\ell(\overline{a}, w)|. \tag{4.10}$$

See Figure 4.1 (case 1) for an illustration for an illustration of the different sets defined above and their relationship, as well as the completion.

Case 2: Now suppose that for voter ℓ, no element in $a \in \overline{a}$ is known to be preferred to w. If $w \in u_\ell(\overline{a})$ then $PMR(u_\ell(\overline{a}), \{w\}, p_\ell) = 0$, since any adversarial completion can place w above all alternatives in $u_\ell(\overline{a}) \setminus \{w\}$ (otherwise regret would be negative). Otherwise the desired completion must maximize the gap between w and any alternative in $u_\ell(\overline{a})$. The following alternatives can be placed between w and $\overline{a}$:

$$B'_\ell(\overline{a}, w) = \{b \in A \setminus \overline{a} : b \not\succ_\ell w \text{ and } \forall a \in u_\ell(\overline{a}), a \not\succ_\ell b\}.$$

The relative ordering of these alternatives does not impact regret. With $B'_\ell(\overline{a}, w)$ placed below w, some alternative from $u_\ell(\overline{a})$ must lie immediately below the last element of this set (becoming the most preferred $a \in \overline{a}$). Thus, we have:

$$PMR(u_\ell(\overline{a}), \{w\}, p_\ell) = \begin{cases} 1 + |B'_\ell(\overline{a}, w)| & \text{if } w \notin u_\ell(\overline{a}), \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

See Figure 4.1 (case 2) for an illustration (where $w \notin u_\ell(\overline{a})$).

In both cases, the undominated sets $u_\ell(\overline{a})$ and $u_\ell(\overline{w})$ can be computed in $O(K^2)$ time.

[2]For "nonlinear" scoring rules, where the score difference for two alternatives depends not just on relative rank position, but also absolute rank position, placement of alternatives above or below a and w requires more care, but is straightforward in most cases.

In case 1, $u_\ell(\overline{a})^+_w$ can be computed in $O(K)$ time once $u_\ell(\overline{a})$ is known, and $B_\ell(\overline{a}, w)$ can be computed in $O(mK)$ time by checking if each $b \in A$ satisfies the constraints with respect to $u_\ell(\overline{a})^+_w$ and w. For case 2, $B'_\ell(\overline{a}, w)$ can be found in time $O(mK)$ by checking each $b \in A$ with w and the alternatives in $u_\ell(\overline{a})$. Using Eqs. 4.9 and 4.8, $PMR(\overline{a}, \overline{w}, \mathbf{p})$ can be computed in $O(n(K^2 + K(K + mK))) = O(nmK^2)$ time. Note that for $K = 1$, the approach is identical to PMR computation for the single-winner Borda rule. Putting this together we have:

Theorem 8. *$PMR(\overline{a}, \overline{w}, \mathbf{p})$ is given by:*

$$\sum_{\ell \in N} \max_{w \in u_\ell(\overline{w})} \begin{cases} -|u_\ell(\overline{a})^+_w| - |B_\ell(\overline{a}, w)| & \text{if } \exists a \in \overline{a} : a \succ_\ell w, \\ 1 + |B'_\ell(\overline{a}, w)| & \text{otherwise, and } w \notin u_\ell(\overline{a}), \\ 0 & \text{otherwise,} \end{cases}$$

and is computable in $O(nmK^2)$ time.

The minimax optimal slate $\overline{a}^*_{\mathbf{p}}$ can be constructed by computing max regret $MR(\overline{a}, \mathbf{p})$ for each slate $\overline{a}$ of size K, and then selecting the slate $\overline{a}^*_{\mathbf{p}}$ that minimizes max regret. In turn, $MR(\overline{a}, \mathbf{p})$ can be computed by determining the PMR of slate $\overline{a}$ for each witness set $\overline{w}$ of size K. Hence:

Proposition 3. *The minimax regret optimal slate $\overline{a}^*_{\mathbf{p}}$ can be computed in time $O(nm^{2K+1}K^2)$.*

The additive decomposability of PMR has the nice computational consequence that, during the course of incremental elicitation (see Section 4.3), one need only update the contributions to PMR of those agents who have their partial preferences updated by responding to a query. For slates of small bounded size K, enumeration of alternative sets may be practical. Indeed, if K is bounded, minimax optimal slates can be computed in polynomial time (in n and m). In other words, the problem is fixed-parameter tractable with respect to slate size K.

However, in general, since this form of proportional representation and budgeted social choice is an NP-hard optimization problem [84] (as are related forms [103]), finding the minimax optimal slate is also NP-hard (simply let $\mathbf{p}$ be a full preference profile). Indeed, even simply computing $MR(\overline{a}, \mathbf{p})$ is NP-hard:

Theorem 9. *Given threshold $r \geq 0$, partial profile $\mathbf{p}$, set size K, and set $\overline{a}$ of size at most K, deciding if $MR(\overline{a}, \mathbf{p}) \geq r$ (i.e., does some set $\overline{w}$ of size at most K satisfy $PMR(\overline{a}, \overline{w}, \mathbf{p}) \geq r$) is NP-complete.*

Proof. This problem belongs in NP as we can check whether a (polynomially sized) certificate $\overline{w}$ has the property that $PMR(\overline{a}, \overline{w}, \mathbf{p}) \geq r$ in polynomial time by Theorem 8. NP-hardness is easy to see using a simple reduction from the limited choice (LC) problem with Borda scoring (see Section 2.3), which is NP-hard [83]. Given any LC instance with budget K and complete profile $\mathbf{v}$ over m alternatives, we transform it into a partial profile $\mathbf{p}$ with $m + K$ alternatives (the m original alternatives plus K "dummy" alternatives). Set $\overline{a}$ to be the dummy alternatives. Define each partial p_ℓ to be identical to v_ℓ on the top-m ranked alternatives, while the rest of the ranking is unspecified. Computing whether some slate $\overline{w}$ of size at most K has $PMR(\overline{a}, \overline{w}, \mathbf{p}) \geq r$ can then be used to determine if there is a slate $\overline{w}'$ with score above a threshold in the LC instance. □

4.2.2 A Greedy Algorithm for Robust Slate Optimization

Given the intractability of computing a minimax optimal slate $\overline{a}^*_{\mathbf{p}}$ for a partial profile $\mathbf{p}$ in general, we investigate the possibility of an efficient algorithm for approximating the problem that will be practical even for large values of K. It turns out that a relatively simple greedy optimization procedure can be used for this purpose.

To develop this greedy approach, we first define the following problem, which we call the *additional alternative problem*. Assume a partial profile $\mathbf{p}$ and a fixed set $\overline{a}$ of $k-1$ alternatives; if one can add a kth alternative to the set, which next alternative minimizes maximum regret under the PR/limited choice model? We define this problem in the obvious way:

$$PMR(a, w, \mathbf{p}|\overline{a}) = PMR(\overline{a} \cup \{w\}, \overline{a} \cup \{a\}, \mathbf{p}) \tag{4.12}$$

$$MR(a, \mathbf{p}|\overline{a}) = \max_{w \in A} PMR(a, w, \mathbf{p}|\overline{a}) \tag{4.13}$$

$$MMR(\mathbf{p}|\overline{a}) = \min_{a \in A} MR(a, \mathbf{p}|\overline{a}) \tag{4.14}$$

$$a^*_{\overline{a},\mathbf{p}} \in \operatorname*{argmin}_{a \in A} MR(a, \mathbf{p}|\overline{a}). \tag{4.15}$$

Here $PMR(a, w, \mathbf{p}|\overline{a})$ denotes the pairwise max regret of extending slate $\overline{a}$ by adding a rather than w, while $MR(a, \mathbf{p}|\overline{a})$ denotes the regret of extending $\overline{a}$ with a rather than some other alternative in the worst-case (over completions of $\mathbf{p}$). Minimax regret and the minimax optimal "extension" of $\overline{a}$, i.e., $a^*_{\overline{a},\mathbf{p}}$, are defined in the obvious way. Note that setting $k = 1$ gives the single-winner robust voting problem addressed earlier.

The additional alternative problem can be solved in polynomial time. We can explicitly compute the pairwise max regret $PMR(a, w, \mathbf{p}|\overline{a})$ of all $m(m-1)/2$ pairs of alternatives (a, w) (where a is a proposed additional alternative and w is an adversarial

witness), using intuitions very similar to those above (as we discuss below). Note that while we can apply our previous algorithm for finding the pairwise max regret for arbitrary pairs of slates $(\overline{a}, \overline{w})$, the algorithm and analysis of $PMR(a, w, \mathbf{p}|\overline{a})$ that we provide below offers a factor k speedup over the direct application of our earlier algorithm.

With PMR in hand, we can readily determine minimax regret using Eqs. 4.13 and 4.14. Hence, we need only show that PMR can be computed in polynomial time. As above, we can compute each voter ℓ's contribution $PMR(a, w, p_\ell|\overline{a})$ independently. We again consider two cases.

Case 1: If $a \succ_\ell w$, then $PMR(a, w, p_\ell|\overline{a}) \leq 0$ since adding w to $\overline{a}$ cannot improve ℓ's score any more than adding a. Assuming $a \succ_\ell w$, if there is some $b \in \overline{a}$ such that $a \not\succ_\ell b$, then b can be ordered over a and $PMR(a, w, p_\ell|\overline{a}) = 0$. However, if $a \succ_\ell b$ for all $b \in \overline{a}$, regret must be negative. $PMR(a, w, p_\ell|\overline{a})$ is then maximized (or negative regret minimized) by placing as few alternatives as possible between a and the best element of $\overline{a} \cup \{w\}$. For any $a \succ_\ell b$, define

$$T_\ell(a, b) = \{b' : a \succ_\ell b' \succ_\ell b\}.$$

Then regret is maximized by ordering the alternatives in $u_\ell(\overline{a} \cup \{w\})$ such that the element with the fewest possible alternatives between it and a is ranked first. This gives:

$$PMR(a, w, p_\ell|\overline{a}) = \max_{b \in u_\ell(\overline{a} \cup \{w\})} -|T_\ell(a, b)| - 1.$$

Case 2: If $a \not\succ_\ell w$, then $PMR(a, w, p_\ell|\overline{a}) \geq 0$. In this case, if there is some $b \in \overline{a}$ such that $b \succ_\ell w$, then w can never be selected; but since w can be ordered over a, $PMR(a, w, p_\ell|\overline{a}) = 0$. However, if there is no $b \in \overline{a}$ with $b \succ_\ell w$, then regret is maximized by maximizing the gap between w and the best element of $\overline{a} \cup \{a\}$. In particular, the alternatives $B'_\ell(\overline{a} \cup \{a\}, w)$, as defined above, can all be ordered between w and the best such alternative. This gives us:

$$PMR(a, w, p_\ell|\overline{a}) = |B'_\ell(\overline{a} \cup \{a\}, w)| + 1.$$

Taken together, this shows:

Theorem 10. *Given a (partial) slate $\overline{a}$, partial vote p_ℓ, and two alternatives $a, w \in A$, pairwise max regret $PMR(a, w, p_\ell|\overline{a})$ for the additional alternative problem can be computed in polynomial time.*

This gives rise to a very simple *greedy algorithm* for approximating a minimax optimal

Algorithm 2 Greedy algorithm

$\overline{a} \leftarrow \emptyset$
for $k = 1$ **to** K **do**
  MMR $\leftarrow \infty$
  for $a \in A$ **do**
    MR $\leftarrow -\infty$
    for $w \in A : w \neq a$ **do**
      for $\ell \in N$ **do**
        if $a \succ_\ell w$ **then**
          PMR $\leftarrow$ PMR$+\max_{b \in u_\ell(\overline{a} \cup \{w\})} -|T_\ell(a, b)| - 1$
        else
          PMR $\leftarrow$ PMR$+|B'_\ell(\overline{a} \cup \{a\}, w)| + 1$
        end if
      end for
      if PMR $>$ MR **then**
        MR $\leftarrow$ PMR
      end if
    end for
    if MR $<$ MMR **then**
      MMR $\leftarrow$ MR
      $a^* \leftarrow a$
    end if
  end for
  $\overline{a} \leftarrow \overline{a} \cup \{a^*\}$.
end for

K-set: starting with the empty slate $\overline{a}_0 = \emptyset$, at each of iteration $k \leq K$ we add alternative $a_k^* = a^*_{\overline{a}_{k-1},\mathbf{p}}$, i.e., the alternative with least max regret given the prior alternatives, to slate $\overline{a}_{k-1}$. The method is detailed in Algorithm 2. While this algorithm comes with no strong approximation guarantees—though we provide some weaker guarantees below—we show in Section 4.4 that it works extremely well in practice.

In terms of run time, case 1 takes $O(mk)$ time (at the kth iteration) and case 2, as discussed previously, takes $O(k^2 + mk) = O(mk)$ time. Computing pairwise max regret for all pairs (a, w), across all agents, and finding the next best alternative $a^*_{\overline{a}_k,\mathbf{p}}$ for each of the K spots on slate $\overline{a}$ results in a total running time of $O(nm^3K^2)$.

There are two reasons the greedy algorithm is not guaranteed to find the minimax optimal slate. The first is unrelated to preference uncertainty: *even with complete preference information*, the greedy algorithm is unable to provide an optimal K-slate in general. In other words, it can produce a slate that has positive max regret. However, the greedy algorithm does provide a $1 - \frac{1}{e}$ approximation in the full information setting. It is not hard to see that if we have sufficient information to make the "optimal greedy choice" at each stage, then the regret-based approach will correspond to the exact greedy algorithm described by Lu and Boutilier [83]:

Proposition 4. *If $MR(a_k^*, \mathbf{p}|\overline{a}_{k-1}) = 0$ for all $k \leq K$, then the greedy-MMR set $\overline{a}_K$ is identical to the set produced by the (full-information) greedy algorithm given any $\mathbf{v} \in \mathsf{C}(\mathbf{p})$.*

Proof. We prove this by induction on the number of greedy-MMR iterations k. In the base case, the full information greedy algorithm will select a single alternative, that with highest score (i.e., the winner under a positional scoring rule). If $MR(a_1^*, \mathbf{p}) = 0$, this implies a_1^* is a necessary winner and hence must have the greatest score, so greedy-MMR selects the score optimal alternative.

Let $\mathbf{v}^*$ be the true but unknown completion of $\mathbf{p}$. Now for some $k < K$, suppose that the following holds of the greedy-MMR slate $\overline{a}_k$: each a_i^* added to the greedy-MMR slate at iteration i, for $i \leq k$, satisfies $a_i^* \in \text{argmax}_{a' \in A} s(\overline{a}_{i-1} \cup \{a'\}, \mathbf{v}^*)$, where $\overline{a}_{i-1} = \{a_1^*, \ldots, a_{i-1}^*\}$. Consider the $k+1$-st iteration: at that point, the chosen alternative a_{k+1}^* has conditional max regret is zero, i.e., $MR(a_{k+1}^*, \mathbf{p}|\overline{a}_k) = 0$. By definition of conditional max regret, any other alternative $a' \in A$ satisfies $s(\overline{a}_{k-1} \cup \{a'\}, \mathbf{v}) \leq s(\overline{a}_k, \mathbf{v})$ for any completion $\mathbf{v} \in \mathsf{C}(\mathbf{p})$. In particular, this holds for $\mathbf{v} = \mathbf{v}^*$, which implies, by definition, that a_{k+1}^* is a full-information greedy optimal choice at iteration $k+1$. □

If the last alternative added has non-zero max-regret, we are assured that true minimax regret is also nonzero:

Observation 4. If $MR(a_K^*, \mathbf{p}|\overline{a}_{K-1}) > 0$, then $MMR(\mathbf{p}) > 0$.

Unfortunately, we cannot be sure that if only the last element has zero regret that we have found the greedy-optimal slate. But even if the minimax-optimal alternative a_i^* does not have zero max regret, we can still obtain bounds on the quality of the solution. We can consider each iteration of our greedy algorithm for adding alternatives to the slate under partial information as *approximating* the corresponding full-information greedy algorithm—it is only able to approximate the choice of alternative at each iteration because it is working with the partial profile $\mathbf{p}$ rather than the true, unknown profile $\mathbf{v}$. However, using known results for bounding the quality of *approximate greedy optimization* [57], we can provide a bound on the quality of the slate $\overline{a}_K^{grd}(\mathbf{p})$ produced by the greedy algorithm under partial information (with respect to the *true profile* $\mathbf{v}$) relative to the true optimal slate $\overline{a}_K^*(\mathbf{v})$ given full information. Specifically:

Proposition 5. *Let* $\mathbf{p}$ *be a (known) partial profile and* $\mathbf{v} \in \mathsf{C}(\mathbf{p})$ *be any (unknown) complete preference profile. Let* $\overline{a}_K^{grd}(\mathbf{p})$ *denote the size* K *slate produced by the regret-based greedy algorithm using partial profile* $\mathbf{p}$*, and* $\overline{a}_K^*(\mathbf{v})$ *be the optimal* K*-slate. Let* m_k *be any lower bound on the marginal value of the* k*th alternative added to a slate by the* full information *greedy algorithm; that is,* $m_k \leq s(\overline{a}_{k-1}^f \cup \{a_k^f\}, \mathbf{v}) - s(\overline{a}_{k-1}^f, \mathbf{v})$ *where* $\overline{a}_{k-1}^f$ *consists of the first* $k-1$ *greedily selected alternatives using the full profile* $\mathbf{v}$*. If* $MR(a_k^p, \mathbf{p}|\overline{a}_{k-1}) \leq \frac{\alpha-1}{\alpha} m_k$ *for all* $k \leq K$*, for some* $\alpha \geq 1$*, where* a_k^p *is the alternative added by the greedy algorithm with partial profile* $\mathbf{p}$*, then*

$$s(\overline{a}_K^{grd}(\mathbf{p}), \mathbf{v}) \geq (1 - \frac{1}{e^{1/\alpha}}) s(\overline{a}_K^*(\mathbf{v}), \mathbf{v}).$$

In other words, the greedy regret-based algorithm constructs a slate that is within a factor of $1 - \frac{1}{e^{1/\alpha}}$ *of the optimal (full-information) slate despite working with incomplete information.*

Proof. We prove this bound by relating the marginal improvement offered at the kth iteration of the greedy algorithm when run using full information $\mathbf{v}$ (call this the full-information greedy algorithm) with that when using conditional max regret to add an alternative to the slate (call this the partial-information algorithm). Let $\overline{a}_{k-1}^p$ denote the slate constructed by the partial-information algorithm at iteration $k-1$. Let a_k^f be the alternative that would be added by full-information greedy to slate $\overline{a}_{k-1}^p$ constructed to that point, and let $FI_k = s(\overline{a}_{k-1}^p \cup \{a_k^f\}, \mathbf{v}) - s(\overline{a}_{k-1}^p, \mathbf{v})$ denote its marginal value. Similarly, let a_k^p denote the alternative added by the partial-information algorithm, and $PI_k = s(\overline{a}_{k-1}^p \cup \{a_k^p\}, \mathbf{v}) - s(\overline{a}_{k-1}^p, \mathbf{v})$ denote its marginal value. Note that we are measuring

marginal value relative to the *true* underlying profile $\mathbf{v}$ (notwithstanding the fact that the partial-information algorithm does not have access to this profile). Finally, let $\overline{a}_K^{grd}(\mathbf{p})$ denote the K-slate produced by the partial-information algorithm and $PI = s(\overline{a}_K^{grd}(\mathbf{p}), \mathbf{v})$ denotes its value under $\mathbf{v}$, and let $OPT = s(\overline{a}_K^*(\mathbf{v}), \mathbf{v})$ be the value of the optimal slate $\overline{a}_K^*(\mathbf{v})$.

At stage k the partial-information algorithm adds an alternative a_k^p that minimizes conditional max regret. By definition of conditional max regret, the marginal value of a_k^p w.r.t. $\overline{a}_{k-1}^p$ must be within an additive factor $MR(a_k^p, \mathbf{p}|\overline{a}_{k-1})$ of the marginal value of a_k^f. Thus we have:

$$FI_k \leq PI_k + MR(a_k^p, \mathbf{p}|\overline{a}_{k-1}) \tag{4.16}$$

$$\leq PI_k + \frac{\alpha - 1}{\alpha} m_k \tag{4.17}$$

$$\leq PI_k + \frac{\alpha - 1}{\alpha} FI_k \tag{4.18}$$

(where the second and third inequalities follow by the statement of the theorem). This implies $\frac{1}{\alpha} FI_k \leq PI_k$; i.e., the partial-information greedy algorithm offers an $\frac{1}{\alpha}$-approximation of the full-information algorithm with respect to the additional alternative problem.

It can be shown that a greedy algorithm applied to a submodular optimization problem in which the alternative added at each step does not maximize marginal improvement, but does provide a γ-approximation of the maximal marginal improvement, provides a $(1 - \frac{1}{e^\gamma})$ approximation to the submodular optimization problem [57]. Since slate optimization for the full information problem is submodular [83], we have

$$PI \geq (1 - \frac{1}{e^{1/\alpha}}) OPT.$$

□

4.3 Preference Elicitation

We now turn our attention to the question of incremental elicitation of voter preferences. When attempting to find an optimal K-slate with a partial preference profile $\mathbf{p}$, we cannot guarantee that an optimal slate can be obtained (regardless of whether we resort to greedy or exact optimization)—specifically, if $MMR(\mathbf{p}) > 0$, no slate can be guaranteed to be optimal. To improve the quality of the slate, further information must be elicited from one or more voters. Our goal is to reduce *relevant* uncertainty, i.e., find those queries that

have the greatest potential to reduce minimax regret. To do this we adapt the principles of the single-winner *current solution strategy (CSS)*, introduced in Section 3.3, to work with slates. We first define CSS in this new context, then provide empirical evaluation of CSS, both in terms of its the number of queries required to reach solutions with low minimax regret, and its running time.

The Current Solution Strategy (CSS)

We focus here on pairwise comparison queries, in which some voter ℓ is asked whether $a \succ_\ell a'$; but the basic principles can be applied to other forms of queries (e.g., top-k queries, as discussed in Section 3.3). The use of CSS differs depending on whether we are using the greedy heuristic or optimal MMR computation to determine our K-slates. We present our approach in the context of greedy MMR computation since it is the more practical method for problems involving slates of reasonable size. The general principles can be readily adapted to optimal MMR computation as well.

Our elicitation scheme works by using the greedy algorithm to compute an (approximately) minimax optimal slate $\overline{a}^*_{\mathbf{p}} = \langle a^*_1, \ldots, a^*_K \rangle$ given the current partial profile $\mathbf{p}$. If $MR(a^*_K, \mathbf{p}|\overline{a}^*_{K-1}) = 0$, we treat this as an (approximately) minimax optimal slate and stop. Otherwise, we know that $MMR(\mathbf{p}) > 0$, so we select a voter ℓ and pairwise comparison query $a \succ_\ell a'$ with the greatest potential to reduce $MR(a^*_K, \mathbf{p}|\overline{a}^*_{K-1})$, using CSS. Let a^*_K be the last alternative added to the slate and w_K be the witness alternative (i.e., where $MR(a^*_K, \mathbf{p}|\overline{a}^*_{K-1}) = PMR(a^*_K, w_K, \mathbf{p}|\overline{a}^*_{K-1})$. CSS identifies the appropriate query (and its potential) for a particular voter i based on several specific cases/sub-cases.

Case 1: Suppose $a^*_K \succ_\ell w_K$. Then ℓ's contribution PMR_ℓ to $PMR(a^*_K, w_K, \mathbf{p}|\overline{a}^*_{K-1})$ must be $PMR_\ell \leq 0$. If $PMR_\ell = 0$, then either: (i) a^*_K is dominated in ℓ's partial order p_ℓ by some $a_j \in \overline{a}^*_{K-1}$, or (ii) a^*_K is not dominated by any such a_j. In case (i), no query can reduce $MR(a^*_K, \mathbf{p}|\overline{a}^*_{K-1})$ since voter ℓ would never select either of a^*_K or w_K given the rest of the slate $\overline{a}^*_{K-1}$, so *no query* is asked of ℓ. In case (ii), the adversary can set $PMR_\ell = 0$ by ordering some alternative $a_j \in \overline{a}^*_{K-1}$ over a^*_K (if no such alternative were possible, PMR_ℓ would have to be negative). In this case, any query that prevents a_j from being orderable above a_k can reduce PMR_ℓ (by making it negative). Specifically, any query of the form $b \succ_\ell c$ for $b \in a^*_K \cup \{a : a^*_K \succ_\ell a\}$ and $c \in a_j \cup \{a : a \succ_\ell a_j\}$ will suffice. Since the degree of PMR_ℓ is determined by the relationship of a^*_K to w_K and *not the gap* with a_j, we choose query $a^*_K \succ_\ell a_j$ since it is implied by any other.

If $PMR_\ell < 0$, then (iii) a^*_K dominates each $a_j \in \overline{a}^*_{K-1}$ as well as w_K. Queries that can extend the advantage over the best alternative of $w^* \in u_\ell(\{w_K\} \cup \overline{a}^*_{K-1})$ (i.e., make

PMR_ℓ even smaller) take two forms: learning that $a^*_K \succ_\ell c$ for some ancestor c of w^* (since we do not know w^*, we choose an arbitrary alternative in $u_\ell(\{w_K\} \cup \overline{a}^*_{K-1})$); or learning that $d \succ_\ell w^*$ for some descendant d of a^*_K. The query with the greatest potential is that with the largest number of descendants (resp. ancestors) lying between it and c (resp. d) in p_ℓ.

Case 2: Suppose $a^*_K \not\succ_\ell w_K$. Then ℓ's contribution PMR_ℓ to $PMR(a^*_K, w_k, \mathbf{p}|\overline{a}^*_{K-1})$ must be $PMR_\ell \geq 0$. (i) If $PMR_\ell = 0$, then we must have w_K dominated by some $a_j \in \overline{a}^*_{K-1}$. In this case we ask no query. (ii) If $PMR_\ell > 0$, then w_K is not dominated by any $a_j \in \overline{a}^*_K$ (i.e., by none of the K alternatives). Then regret can be reduced only by asking a query that removes elements from the set $B'_\ell(\overline{a}^*_K, w_K)$ by either placing w_K (or one of its ancestors) below some a_j (it is only necessary to consider those in $u_\ell(\overline{a}^*_K)$); by placing some $a_j \in u_\ell(\overline{a}^*_K)$ (or one of its descendants) above w_K; or placing some element that is incomparable to both a^*_K and w_K either above w_K (hence placing its ancestors above as well) or below a^*_K (hence placing its descendants below as well). In the case that w_K dominates all of $\overline{a}^*_K$, one can ask queries that either (a) move a descendant d of w_K, where d is not an ancestor of some $a_j \in \overline{a}^*_K$, below such an a_j; or (b) move an ancestor c of some a_j, where c is not a descendent of w_K, above w_K. The potential of a query to reduce PMR_ℓ is measured by the number of elements it removes from the set $B'_\ell(\overline{a}^*_K, w_K)$.

4.4 Empirical Evaluation

We now describe experiments designed to test the ability of our greedy slate optimization method, when coupled with the CSS elicitation strategy, to find good (or even optimal) slates of alternatives with few voter queries. We evaluate the approach on two real datasets as well as on more systematically generated random data. As in single-winner problems, we use the *Sushi* data set and the *Irish* dataset, and draw profiles of 100 random users/voters from these sets as described in Section 3.4). In the Sushi experiments we set the slate size to be $K = 3$, while in the Irish experiments we set $K = 4$. We also test CSS by generating random profiles of 100 voter rankings over 10 alternatives, with voter rankings drawn i.i.d. from a Mallows distribution, again using the same methodology as in Section 3.4. In the *Mallows* experiments, we set $K = 3$ and analyze elicitation performance as we vary the dispersion parameter ϕ.

Experimentally, each instance consists of a full profile. We start each run with no voter preference information, then, using CSS to generate queries, elicit pairwise comparisons from voters (who respond accurately based on their underlying preferences). After each

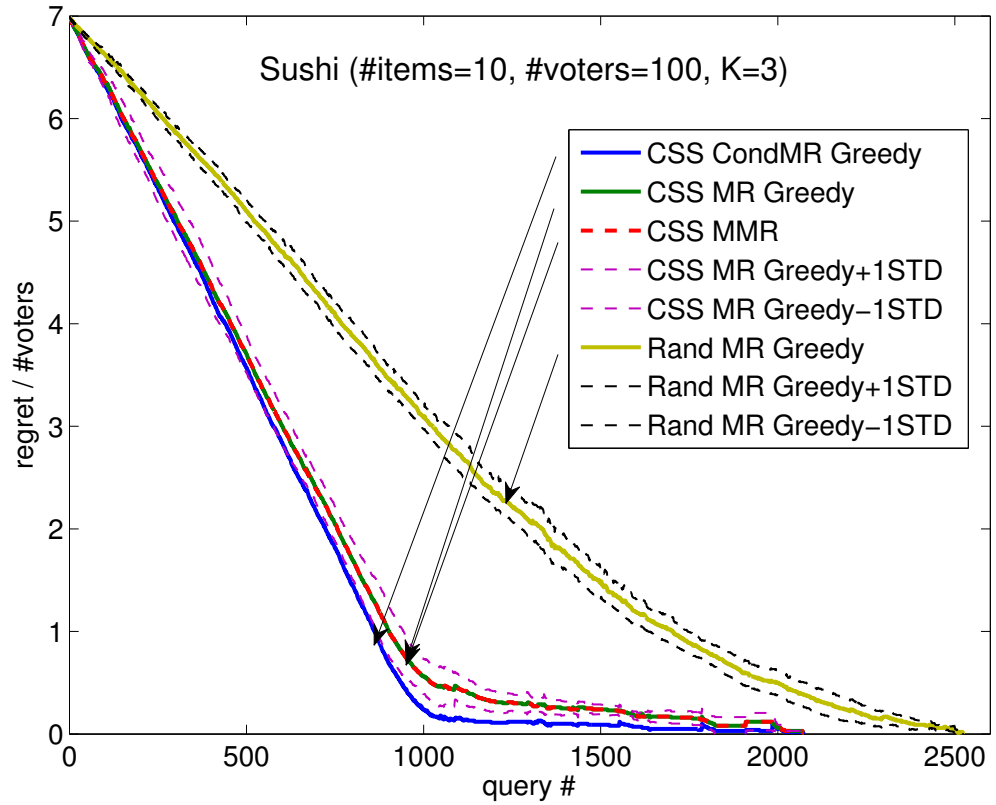

Figure 4.2: Performance of Greedy/CSS on Sushi (20 trials).

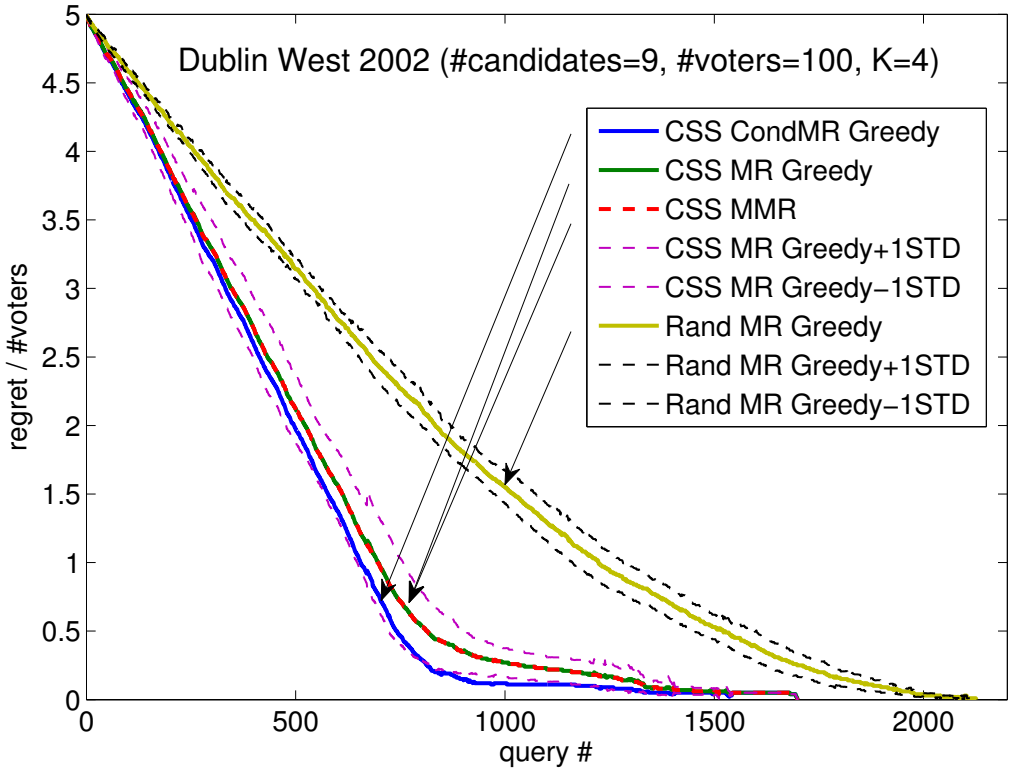

Figure 4.3: Performance of Greedy/CSS on Irish (20 trials).

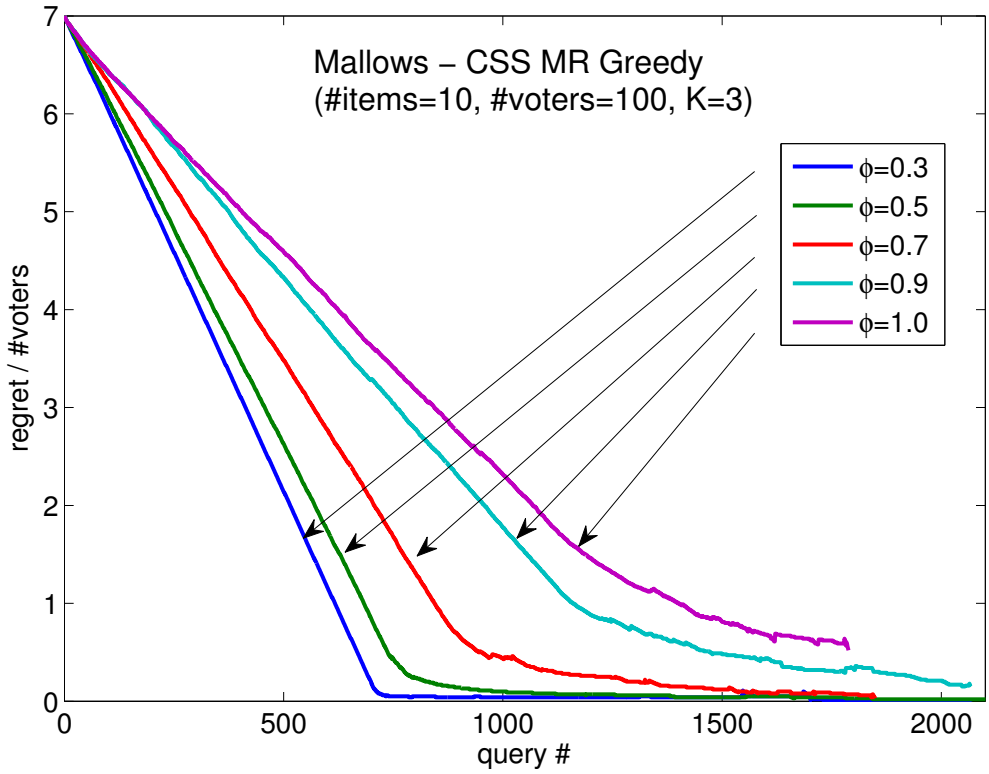

Figure 4.4: Performance of Greedy/CSS on Mallows (20 trials).

query/round, we use the greedy algorithm to compute an (approximately) optimal K-slate. Elicitation terminates once the conditional max regret (CondMR) of the Kth alternative added to the slate, $MR(a^*_K, \mathbf{p}|\overline{a}^*_{K-1})$, is zero. (Note that after each elicitation round a "fresh" slate is constructed; elicitation is terminated at that round if the Kth alternative added to the slate *at that round* has zero conditional max regret.) We also compare this to the use of exact minimax regret computation at each round of elicitation to determine the truly optimal K-slate.

Results in Figures 4.2 and 4.3 show performance for the *Sushi* and *Irish* datasets. The plots show, for the slate produced by the Greedy algorithm after each query:

- its conditional max regret (CondMR), i.e., that of the last element added to the slate;
- its true max regret;
- and one sample standard deviation above and below (dotted lines) its regret values (results are averaged over 20 randomly drawn profiles).

We also show the performance of CSS when exact MMR is computed and the optimal slate is generated at each iteration. Finally, we compare CSS with a baseline random strategy

m	$K=2$	$K=3$	$K=4$	$K=6$	$K=8$
10	0.015	0.020	0.023	0.028	0.033
20	0.105	0.152	0.194	0.275	0.345
30	0.342	0.508	0.642	0.987	1.282
50	1.577	2.042	2.247	4.439	6.344

Table 4.1: Average Greedy runtime (sec.), on random Mallows profiles.

that randomly picks a voter and pairwise comparison query (ensuring this response to this query is not implied by that voter's partial ranking, i.e., it is not contained in the transitive closure of previous responses), using Greedy to compute the slate at each round and measuring its max regret. Again, a range of one standard deviation is shown.

These plots indicate that that CSS works very well. It finds a slate with zero max regret "per voter" with only about 20 queries per user in Sushi (resp., 15 in Irish), even with the relatively large ratio of K to m in each setting (30% and 44%, respectively). CSS also reaches near-zero regret in about 10 queries (resp., 8) per user; thus, its *anytime profile* is very encouraging for settings where approximately optimal solutions are permissible, especially if approximation provides a significant reduction in elicitation burden. Note that the true regret may be zero even if max regret is not. We contrast the number of queries needed by CSS with the demands of complete sorting to provide a full ranking, which requires $\approx m \log_2(m)$ pairwise comparisons using methods with good average case performance, or equivalently 34 (resp., 29) queries. Random requires 25 (resp., 22) queries per user to reach zero regret, and has a much worse anytime profile.

Notice that the greedy algorithm itself works extremely well: it almost always finds the minimax optimal slate—the MR Greedy and MMR curves coincide almost exactly—and in the rare cases that it does not, Greedy MR is very close to true MMR. MR may not decrease monotonically, as preference updates may "mislead" Greedy into choosing an inferior slate (by contrast, true MMR is non-increasing). CondMR is also a good proxy for true max regret: in *Sushi*, the per-voter difference is at most 0.41 and in *Irish* at most 0.24. Thus, CondMR—which can be computed efficiently—is an excellent surrogate for MR—which is NP-hard—as a quality measure and a stopping criterion for elicitation.

Mallows results in Figure 4.4 show how the same quantities change as a function of the total number of queries, for different dispersion values ϕ. The results show that, unsurprisingly, more concentrated preference distributions (smaller ϕ) require fewer queries to find good slates. This is consistent with results for in single winner voting (see Section 3.4).

Table 4.1 shows wall clock runtimes for Greedy with different values of m (alterna-

tives) and K (slate size) on a 3.0GHz Intel Xeon processor. Results are averaged over "complete" CSS elicitation runs (i.e., over all optimizations solved until elicitation reaches a slate whose CondMR is zero), on random profiles of $n = 40$ voters drawn from a Mallows distribution with $\phi = .7$. Average runtime increases significantly with the number of candidates m, but less dramatically with K. This is consistent with the "quadratic in K" and "cubic in m" computational analysis described above. Still, Greedy is very practical, taking only 6.3s. to find optimal slates for $m = 50$ candidates and slates of size $K = 8$.

4.5 Conclusion

In multi-winner settings, we extended our approach to deal with *proportional representation* schemes. We developed algorithms for both the exact and greedy optimization of slates with respect to minimax regret. The greedy method is especially practical from a computational perspective. We provided some theoretical bounds on the performance of the greedy method, but more importantly demonstrated that it often outputs slates that are either minimax optimal or very close to optimal. We also adapted the CSS elicitation heuristic and showed that, when coupled with the greedy slate algorithm (even when using conditional max regret as surrogate for max regret), it finds very good slates while asking for relatively little preference information. As with single-winner CSS, it has a desirable anytime profile.

For future work, we would like to apply the robust optimization and elicitation framework to more multi-winner voting rules. This includes natural rules based on positional scoring functions, such as selecting a slate of alternatives with the highest number of approval votes, or taking the top-K candidates ranked by their Borda scores, etc. Such rules would have a scoring function that is additive with respect to the slates' alternatives. Determining the optimal slate with a full preference profile with respect to these voting rules is computational easy. However, algorithms for computing the minimax optimal slate might be computational intractable and more involved than the single-winner case. Nevertheless, developing heuristics, such as greedy algorithms, would be of practical value. Effective elicitation algorithms may also be subtly different from the CSS strategy developed in this chapter. As the score of a slate would be additive (as opposed to a max over each alternative's score within the slate), the elicitor may require more preference information before reaching a slate of the same MMR quality.

Another direction is to incorporate probabilistic information of agent preferences to improve the quality or confidence of the recommended slate. Many multi-winner problems

appear in recommender systems, and therefore are of practical interest in low-stakes domains. Because MMR provides robust, worst guarantees, in low-stakes settings one may not need such stringent guarantees. Using probabilistic models of preferences may help in the selection of slates that are of high quality/score in expectation (or are of low regret in expectation). The use of such side information can not only improve estimates of the true quality of a slate, but also reduce the amount of information that must be elicited from the agents. Such probabilistic models can also be of use when MMR is the required solution concept. One can use preference models to guide in the elicitation of preferences. Such probabilistic information may help identify particular agents that are of a certain *type*, then elicitation can proceed to query an agent who would reveal information that reduces MMR the most in expectation. Before we can use such preference models, we need to devise algorithms that can *learn* these models from data. This is the topic of the following chapter.

Chapter 5

Learning Rankings with Pairwise Preferences

In this chapter, we focus on developing algorithms for learning *mixture models* to fit distributions over rankings—using only pairwise comparison data—a technical challenge that had been previously unsolved. Having these learned models allows us to tackle a variety of social choice problems including elicitation, recommendation, prediction, vote manipulation and the design of optimal social choice rules.

Many existing learning and inference methods impose restrictive assumptions on the form of user preferences that can be admitted as evidence. We develop the first algorithms for learning Mallows models (and mixtures thereof) from pairwise comparison data—the fundamental building blocks of partial orders. At the heart of our technique is a new algorithm, the *generalized repeated insertion model (GRIM)*, which allows sampling from arbitrary ranking distributions, and conditional Mallows models in particular. While we show that sampling from a Mallows model with pairwise evidence is computationally difficult in general, we develop approximate samplers that are exact for many important special cases—and have provable bounds with pairwise evidence—and derive algorithms for evaluating log-likelihood, learning Mallows mixtures, and non-parametric estimation. Experiments on real-world datasets demonstrate the effectiveness of our approach.

The remainder of this chapter is organized as follows. In Section 5.1 we provide some background and motivation for the preference learning problem. In Section 5.2 we describe the necessary preliminaries and discuss related work on learning probabilistic preference models. We introduce our main technical tool, the *generalized repeated insertion method (GRIM)*, in Section 5.4. We show how it can be used to sample from Mallows mixtures conditioned on incomplete preferences by first defining an approximate, but *direct* sampler AMP that is exact for important special cases, and analyzing its com-

putational and statistical properties. We then develop Metropolis and Gibbs sampling methods that exploit AMP to soundly sample any Mallows or Mallows mixture posterior. In Section 5.5 we develop an EM algorithm for learning a Mallows mixture from arbitrary pairwise comparison data that leverages our sampling algorithms, and provide experimental results of this procedure on several real-world data sets in Section 5.6. Section 5.7 extends the framework of Lebanon and Mao [79] for non-parametric estimation to handle evidence in the form of arbitrary ordinal preferences. We conclude in Section 5.8 with a discussion of future directions.

5.1 Motivation

With the abundance of preference data from search engines, review websites, etc., there is tremendous demand for learning detailed models of user preferences to support personalized recommendation, information retrieval, social choice, and other applications. Much work has focused on ordinal preference models and learning user or group rankings of alternatives. Within this setting, we can distinguish two classes of models. First, we may wish to learn an underlying *objective* (or "correct") ranking from noisy data or noisy expressions of user preferences (such as in web search, where user selection suggests relevance), a view adopted frequently in IR and "learning to rank" [21] and occasionally in social choice [126]. Second, we might assume that users have different *types* with inherently distinct preferences, and learn a population model that explains this diversity.

Learning preference types (e.g., by segmenting or clustering the population) is key to effective personalization and preference elicitation in recommender systems, social choice, and numerous other domains. For example, with a learned population preference distribution, choice data obtained from a specific user allows inferences to be drawn about her unobserved preferences, or preference queries to identified that drive down minimax regret. In this work, we focus on the latter setting, learning preference distributions when users have genuinely distinct preferences.

Considerable work in machine learning has exploited ranking models developed in the statistics and psychometrics literature, such as the Mallows model [87], the Plackett-Luce model [101, 85], and others [88], as well as their non-parametric representations [79]. However, most prior research provides methods for learning preference distributions using very restricted forms of evidence about individual user preferences, whether passively observed or actively elicited, ranging from complete rankings, to top-t or bottom-t alternatives, to *partitioned preferences* [79]. Missing from this list are (sets of) arbitrary *pairwise comparisons* of the form "alternative a is preferred to alternative b." Such

pairwise preferences form the building blocks of almost all reasonable evidence about preferences, and subsumes the most general evidential models proposed in the literature. For instance, top-k preferences (e.g., if a, b, c are the top-3 alternatives) can be represented by $m-1$ pairwise comparisons: $k-1$ comparisons represent the relative ranking of the top-k alternatives while $m-k$ comparisons are needed for indicating the preference of the k-th ranked alternative to all other $m-k$ alternatives (e.g., $a \succ b, b \succ c, c \succ d, c \succ e, \ldots$).[1]

Furthermore, preferences in this form naturally arise in active elicitation of user preferences and choice contexts (e.g., web search, product comparison, advertisement clicks), where a user selects one alternative over others in some set [81]. In general, data about a user's preferences will often take the form of arbitrary choice sets as is common in web search, online advertising, product comparison, etc. But none of the prior techniques and algorithms developed can learn from such choice sets. These preferences can be as simple as a single pairwise comparison: "I like alternative a better than b," or as complex as a set of comparisons: "I like a better than $b, c, \ldots$, *and* I like z better than $y, x, \ldots$" In this sense, pairwise comparisons should be viewed as the fundamental building block of ordinal preference ranking. Of course, ordinal preferences do not capture strength of preference; but real-valued or scaled preferences (e.g., movie or book ratings) can be converted to pairwise preferences readily, albeit with some loss of information.

While learning with pairwise preferences is clearly of great importance, it is widely believed that learning probabilistic models of ordinal preference using pairwise comparison data is impractically difficult (indeed, we show this formally in this chapter). As a consequence, the Mallows model is often not used in favour of more inference-friendly models (e.g., Plackett-Luce, which accommodates more general, but still restrictive, preferences [26, 58]). To date, no methods have been proposed for learning from arbitrary collections of consistent pairwise preferences in any of the commonly used ranking models in machine learning. We tackle this problem directly by developing techniques for learning Mallows models, and mixtures thereof, from pairwise preference data.

Our core contribution is the *generalized repeated insertion model (GRIM)*, a new method for sampling from *arbitrary ranking distributions*—including conditional Mallows—that generalizes the repeated insertion method for unconditional sampling of Mallows models [43]. We show that even evaluating the log-likelihood under a Mallows model with respect to arbitrary ordinal data is #P-hard, implying that learning will be at least as difficult. However, we derive another method, which we call AMP, which efficiently,

[1] One exception to this is information about preferences that involve "disjunctive" constraints. For instance, a response to the question "What alternative is ranked t^{th}?" cannot be mapped to a set of pairwise preferences unless the positions t are queried in ascending or descending order (hence inducing top-t or bottom-t preferences).

though approximately, samples from any conditional Mallows distribution given arbitrary pairwise evidence. Moreover, we show that AMP is *exact* for important classes of evidence (including partitioned preferences), and that empirically it provides very close approximations given general pairwise evidence. We use this sampler as the core of a *Monte Carlo* Expectation Maximization (EM) algorithm to learn Mallows mixtures, evaluate log-likelihood, and make predictions about missing preferences. We also extend the non-parametric framework of Lebanon and Mao [79] to handle unrestricted ordinal preference data. Experiments show our algorithms can effectively learn Mallows mixtures, with reasonable running time, on datasets with hundreds of alternatives and thousands of users. Our sampling algorithm can be adapted rather easily to other models as well (e.g., we show how a simple modification allows sampling from Mallows models with a weighted Kendall-tau metric).

5.2 Preliminaries

We begin by describing the ordinal preferences (rankings) used in the work, providing a brief overview of several common probabilistic preference models, with an emphasis on the Mallows ϕ-model (and mixtures). We then outline the repeated insertion model (RIM) [43] for sampling preferences from a Mallows distribution (and draw connections to older models for sampling rankings proposed by Condorcet, Kemeny and Young). We also briefly discuss related work on learning probabilistic preference models.

5.2.1 Ordinal Preferences

For an overview of preference relations and partial preferences, see Section 2.1. In this chapter, we assume that we generally do not have access to the complete preferences of agents, but only partial information about their rankings (e.g., based on choice behavior, query responses, etc.). We assume this data has a very general form: for each agent ℓ we have a set of *revealed pairwise preference comparisons* over A, or simply *preferences*:

$$v_\ell = \{x_1^\ell \succ_\ell y_1^\ell, \ldots, x_{k_\ell}^\ell \succ_\ell y_{k_\ell}^\ell\}.$$

Intuitively, these reflect information about ℓ's preferences revealed by some process. For example, this could represent product-ratings data; preference revealed by selection or purchase of certain items (e.g., web links, products) over others, or responses to survey data. It could also represent information elicited using a process similar to what is described in Chapter 3.

We assume that the transitive closure of v_ℓ is acyclic, in other words, the pairwise comparisons do not collectively contradict one another. Many of the concepts for probabilistic modeling, inference and learning developed in this chapter can be applied *mutatis mutandis* to models where revealed preferences are noisy; however, we leave this topic to future research. Recall that preferences v_ℓ are *complete* if and only if $\mathsf{tc}(v)$ is a total order on A. We let $\mathsf{C}(v)$ denote the set of complete preferences (rankings) that are (consistent) extensions of v, and we let C denote the set of all rankings for some fixed number of alternatives m. We assume that the observed data from agents or users used for inference and learning purposes form a partial preference profile.

We briefly recall the discussion in Section 2.1.3 on distances over preferences. Given ranking $\sigma = \sigma_1\sigma_2\cdots\sigma_m$ and preference v, we denote by $d(v,\sigma)$ the *disagreement* between the two preferences. This is an extension of the Kendall-tau distance that counts the number of pairwise misorderings with respect to $\mathsf{tc}(v)$ and σ. That is, dissimilarity between a partial preference and a ranking is the number of pairwise disagreements among the relative ranking of alternatives, i.e., those pairs in v that are misordered relative to σ. If v is a complete ranking, $d(v,\sigma)$ is the classic *Kendall-tau* metric on rankings. Likewise, $s(v,\sigma)$ denotes the number of pairwise agreements with respect to $\mathsf{tc}(v)$ and σ.

Arbitrary sets v of pairwise comparisons can be used to model a wide range of realistic revealed preferences:

- Complete rankings require $m-1$ pairwise comparisons (e.g, $a_1 \succ a_2$, $a_2 \succ a_3$, $\ldots, a_{m-1} \succ a_m$), and can be elicited with at most $m(m-1)/2$ pairwise comparison queries.
- Top-k preferences [24] require that users provide a complete ranking of their top k most preferred alternatives. These can be represented using $m-1$ pairs: $k-1$ comparisons to order the top k alternatives, and $m-k$ pairs to ensure the kth alternative is ranked above the remaining $m-k$ alternatives. Bottom-k preferences are similar.
- Complete rankings of *subsets* $X \subseteq A$ [58, 26] are also representable in the obvious fashion (requiring $k-1$ comparisons if $|X| = k$).
- Preferences revealed by the choice of an alternative a from $X \subseteq A$ [81] can also be represented using $k-1$ pairs of the form $a \succ b$ for each $b \in X \setminus \{a\}$ (where $|X| = k$). Sets of such choices are captured in the obvious way.
- Ordinal ratings data: if alternatives are scored on an ordinal scale s (e.g., a scale of 1–5 where 5 is most preferred), we simply include $a \succ b$ whenever $s(a) > s(b)$,

assuming that alternatives with the same rating cannot be compared using the level of granularity provided.

Much of the existing work in learning or modelling distributions over ordinal preferences restricts the class of representable preferences. Much work has focused on top-k preferences [24, 90, 55, 47, 48], and its generalizations [79]; other papers have worked with rankings of a subset of alternatives [58, 26]. The main issue in allowing arbitrary consistent collections of pairwise preferences, which can represent all of the above special cases, is the difficult inference problem that results. The primary aim of this chapter is to develop tractable inference algorithms for a much broader and realistic class of preferences. Before closing our discussion of ordinal preferences, we define a recently studied and relatively expressive class of preferences

Definition 18 (Lebanon and Mao [79])**.** A partial preference v is a *partitioned preference* if A can be partitioned into subsets $A_1, \ldots, A_q$ such that:

(a) for all $i < j \leq q$, if $x \in A_i$ and $y \in A_j$ then $x \succ_{\mathsf{tc}(v)} y$; and

(b) for each $i \leq q$, alternatives in A_i are incomparable under $\mathsf{tc}(v)$.

Partitioned preferences are quite general, subsuming some of the special cases above, including top-t or bottom-t preferences, or ratings data. However, they cannot represent many naturally occurring preferences, including those as simple as a single pairwise comparison $a \succ b$. We demonstrate in this chapter that our techniques can be applied effectively to such preferences.

5.2.2 Mallows Models and Sampling Procedures

There are many distributional models of rankings that have been developed in psychometrics, statistics and econometrics to explain choice behavior (Marden [88] provides a good overview). Two of the more popular in the machine learning community are the *Mallows model* [87] and the *Plackett-Luce model* [101, 85]. We focus on Mallows in this work, though we believe our methods can also be extended to other models.

5.2.2.1 The Mallows Model

The *Mallows ϕ-model* (which we simply call the Mallows model hereafter) is typical of a wide-range of distance-based ranking models [87, 88]. As above, let d be the Kendall-tau

distance. The Mallows model is parameterized by a modal or *reference ranking* σ and a *dispersion parameter* $\phi \in (0, 1]$. For any ranking r, the Mallows model specifies:

$$P(r) = P(r \mid \sigma, \phi) = \frac{1}{Z}\phi^{d(r,\sigma)} , \tag{5.1}$$

where $Z = \sum_{r' \in \mathsf{C}} \phi^{d(r',\sigma)}$ is the normalization constant. Note the use of the conditional probability notation, this is only for readability purposes with regards to the distributional parameters, and we assume in this chapter that no prior is placed on model parameters. It can be shown that

$$Z = 1 \cdot (1 + \phi) \cdot (1 + \phi + \phi^2) \cdots (1 + \cdots + \phi^{m-1}). \tag{5.2}$$

When $\phi = 1$ we obtain the uniform distribution over C (in the social choice literature, this model is known as *impartial culture*). As $\phi \to 0$, the distribution concentrates all mass on σ. The model can also be expressed as $P(r|\sigma, \lambda) = \frac{1}{Z}\exp(-\lambda d(r, \sigma))$, where $\lambda = -\ln \phi \geq 0$. Various extensions and generalizations of this model have been developed (e.g., using other distance measures) [88].

5.2.2.2 Condorcet's Decision Problem

We describe a simple sampling procedure proposed by Mallows, Condorcet and further analyzed by Young, since this will motivate the RIM sampler discussed in Section 5.2.2.3. Mallows [87] explained his model using process in which a judge assesses alternatives by repeatedly making pairwise comparisons. The outcome of such a comparison is stochastic and depends on the reference ranking σ. If x and y are compared and x is preferred to y in σ, then the judge "correctly" assesses $x \succ y$ with probability $1 - p_{xy}$, and erroneously assesses $y \succ x$ with probability $p_{xy} < 1/2$. Each assessment is independent of other comparisons. Mallows' process generated a pairwise comparison for each pair of alternatives as described: after all paired comparisons are made, if the result is consistent (i.e., corresponds to a ranking), it is accepted; otherwise the process is repeated. While the error probability p_{xy} can depend in a fairly general way on their positions in σ, if $p_{xy} = p$ for all x, y then we obtain the Mallows model.

Such a probabilistic view of rankings was studied two centuries earlier by Nicolas de Condorcet in the context of collective political decision making [30]. He modeled his view of the role of government, that of making the "right decisions," by considering the selection from a set of choices (e.g., policies), one that maximizes benefit to society. Members of society, or *voters*, express their opinion in the form of a ranking over choices. He

assumed that some (latent) *objective ranking* orders choices from most to least beneficial to society and that each voter is able to provide an independent, random assessment of relative rank of any pair of choices: if $a \succ b$, in the objective ranking a voter will assess that to be the case with probability $1 - p$, with a error probability less than $1/2$. Instead of studying the probabilistic model *per se*, Condorcet addressed the *decision problem*: how to find the ranking most likely to be correct. For the case of three alternatives, he proved that the ranking which minimized the total number of pairwise preference disagreements (i.e., Kendall-tau distance) with respect to the stated voter rankings was the most likely to be correct.

In modern parlance, Condorcet showed how to compute the maximum likelihood estimator (MLE) of the objective or reference ranking. Kemeny [71] proposed the *Kemeny ranking* as a general method for aggregating noisy voter rankings, extending Condorcet's approach to accommodate any number of alternatives. The Kemeny ranking is that which minimizes the total number of pairwise preference disagreements with the set of voter rankings, which Kemeny justified axiomatically (showing it to be the only aggregate ranking that satisfies certain intuitive axioms). A statistical rationale for Kemeny's approach was provided by Young [126], who extended Condorcet's analysis, showing that, for *any* number of alternatives, under Condorcet's noise model, the MLE of the reference ranking is in fact the Kemeny ranking. These two independent threads (Condorcet-Kemeny-Young and Mallows) can both be viewed as statistical estimation of a noisy ranking model. We tie these threads together, showing that Condorcet's noise model for any number of alternatives corresponds to the Mallows models (which implies, by Young's result, that the Kemeny ranking is the MLE for the Mallows model). The Condorcet-Mallows noisy ranking process can be formalized as follows:

Pairwise Comparison Sampling of Mallows

1. Let σ be the reference ranking and $0 \leq p \leq 1/2$.
2. Initialize $v \leftarrow \emptyset$.
3. For each pair of items x, y in A, such that $x \succ_\sigma y$,
   (a) with probability $1 - p$ add $x \succ y$ to v,
   (b) otherwise add $y \succ x$ to v.
4. If v is intransitive, go back to step 1 and start over.
5. v is transitive and corresponds to a ranking.

This pairwise comparison process generates rankings in accordance with the Mallows model (Eq. 5.1), a fact shown by Mallows [87], but which we derive here (since it will be

instructive below). Consider the following distribution over rankings v.

$$P'(v \mid \sigma, p) = \frac{1}{Z'} \prod_{\{x,y\} \subseteq A} \begin{cases} p & \text{if } v \text{ and } \sigma \text{ disagree on } x, y \\ 1-p & \text{otherwise,} \end{cases} \tag{5.3}$$

where Z' is the normalization constant (i.e., the sum of the probabilities generated by the above procedure, over all transitive, complete preferences). The form of this distribution corresponds exactly to the rankings generated. This can be seen by noticing that the generating procedure independently decides for each pair of alternatives x, y, with a flip of p-biased coin, whether to order them according to σ. Since intransitive preferences v are discarded by the procedure, the generating procedure corresponds to P'. We can simplify the expression for P' to:

$$\begin{aligned} P'(v \mid \sigma, p) &= \frac{1}{Z'} p^{d(v,\sigma)} (1-p)^{s(v,\sigma)} \\ &= \frac{1}{Z'} p^{d(v,\sigma)} (1-p)^{\binom{m}{2} - d(v,\sigma)} \\ &= \frac{1}{Z'} (1-p)^{\binom{m}{2}} \left(\frac{p}{1-p} \right)^{d(v,\sigma)} \end{aligned} \tag{5.4}$$

By setting $\phi = \frac{p}{1-p}$, recalling the definition of Z (Eq. 5.2), and noticing that

$$Z' = (1-p)^{\binom{m}{2}} Z \tag{5.5}$$

$$= (1-p)^{\binom{m}{2}} \left(1 + \frac{p}{1-p}\right) \left(1 + \frac{p}{1-p} + \left(\frac{p}{1-p}\right)^2\right) \cdots \tag{5.6}$$

$$\left(1 + \cdots + \left(\frac{p}{1-p}\right)^{m-1}\right), \tag{5.7}$$

we obtain Eq. 5.1. The log-likelihood, given observed complete rankings $r_1, \ldots, r_n$, is

$$\sum_{\ell=1}^{n} \left[d(r_\ell, \sigma) \ln \phi - \ln Z \right].$$

Hence, the MLE ranking is the minimizer of $\sum_{\ell=1}^{n} d(r_\ell, \sigma)$, namely, the Kemeny ranking.

5.2.2.3 The Repeated Insertion Model

The Condorcet/Mallows sampling procedure for drawing rankings from the Mallows distribution can be very inefficient, since it relies on rejection of partially constructed rank-

ings as soon as a single circular triad ($a \succ b \succ c \succ a$) is drawn. While the original motivation for these models was not computational, efficient sampling is important for a variety of inference and learning tasks. Doignon et al. [43] introduce the *repeated insertion model (RIM)* for the analysis of probabilistic models of approval voting, but which also provides a much more effective means of sampling from a Mallows distribution.

RIM is a generative process that gives rise to a family of distributions over rankings and provides a practical way to sample rankings from a Mallows model. The model assumes some reference ranking $\sigma = \sigma_1\sigma_2 \cdots \sigma_m$, and *insertion probabilities* p_{ij} for each $i \leq m, j \leq i$. RIM generates a new *output ranking* using the following process, proceeding in m steps. At step 1, σ_1 is added to the output ranking. At step 2, σ_2 is inserted above σ_1 with probability $p_{2,1}$ and inserted below with probability $p_{2,2} = 1 - p_{2,1}$. More generally, at the ith step, the output ranking will be an ordering of $\sigma_1, \ldots, \sigma_{i-1}$ and σ_i will be inserted at rank $j \leq i$ with probability p_{ij}. Critically, the insertion probabilities are *independent of the ordering of the previously inserted alternatives.*

It is easy to see that one can generate any ranking with the appropriate insertion positions. As we describe below, Doignon et al. [43] show that one can sample from a Mallows distribution using RIM with appropriate insertion probabilities. We now introduce several concepts that can be used to more easily formalize and analyze RIM, and our subsequent extensions of it.

Definition 19. Let $\sigma = \sigma_1 \cdots \sigma_m$ be a reference ranking. Let an *insertion vector* be any positive integer vector $\mathbf{j} = (j_1, \ldots, j_m)$ satisfying $j_i \leq i, \forall i \leq m$; and let I be the set of such insertion vectors. A *repeated insertion function* $\Phi_\sigma : I \to \mathsf{C}$ maps an insertion vector $\mathbf{j}$ into a ranking $\Phi_\sigma(\mathbf{j})$ by placing each σ_i, in turn, into rank j_i, for all $i \leq m$.

This definition is best illustrated with an example.

Example 2. Consider the insertion vector $(1, 1, 2, 3)$ and reference ranking $\sigma = abcd$. In this case, $\Phi_\sigma(1, 1, 2, 3) = bcda$ because:

1. we first insert a into rank 1 and obtain ranking a,
2. we then insert b into rank 1, shifting a down to obtain partial ranking ba,
3. we then insert c into rank 2, leaving b in place, but moving a down, obtaining ranking bca;
4. finally, we insert d at rank 3, giving $bcda$.

By the same process we obtain $\Phi_\sigma(1, 2, 3, 4) = abcd$, and $\Phi_\sigma(1, 1, 1, 1) = dcba$.

Given reference ranking σ, there is a one-to-one correspondence between rankings and insertion vectors.

Observation 5. For any reference ranking σ, the repeated insertion function Φ_σ is a bijection between I and C.

Sampling using RIM can characterized as follows:

Definition 20. The *repeated insertion model* is a probabilistic model over rankings defined by

- A reference ranking σ
- The repeated insertion function $\Phi_\sigma(j_1, \ldots, j_m)$
- A sequence of *insertion probabilities* p_{ij_i} for $i \leq m$, $j_i \leq i$, such that

$$\sum_{j=1}^{i} p_{ij} = 1, \quad \forall i \leq m$$

A ranking is generated at random by first drawing an insertion vector $\mathbf{j} = (j_1, \ldots, j_m) \in I$, where each j_i is drawn independently with probability p_{ij_i}, and then applying the insertion function $\Phi_\sigma(\mathbf{j})$.

Let $\Phi_\sigma^{-1}(r) = (j'_1, \ldots, j'_m)$. Then the probability of generating a particular ranking r under RIM is $\prod_{i \leq m} p_{ij'_i}$. It is easy to see that the Kendall-tau distance between the reference ranking and the ranking induced by an insertion vector is the sum of the number "insertion misorderings" over all alternatives:

Proposition 6. *For any insertion vector* $\mathbf{j} = (j_1, \ldots, j_m) \in I$*, we have that*

$$\sum_{i=1}^{m} i - j_i = d(\Phi_\sigma(\mathbf{j}), \sigma). \tag{5.8}$$

Proof. Observe that whenever σ_i is inserted at the j_ith position, it creates $i - j_i$ pairwise misorderings with respect to alternatives $\sigma_1, \ldots, \sigma_{i-1}$. All pairwise misorderings can be accounted for this way. Summing over all $i \leq m$ gives the Kendall-tau distance. □

Doignon et al. [43] show that by setting the insertion probabilities p_{ij} appropriately, the resulting generative process corresponds to the Mallows model. We reprove their Theorem here, since the proof will be instructive later.

Theorem 11 (Doignon et al. [43]). *By setting insertion probabilities*

$$p_{ij} = \frac{\phi^{i-j}}{(1 + \phi + \cdots + \phi^{i-1})},$$

for $j \leq i \leq m$, the distribution induced by RIM with insertion function Φ_σ is identical to that of the Mallows model with reference ranking σ and dispersion parameter ϕ.

Proof. We reprove the Doignon et al. [43] theorem. Let r be any ranking and σ the reference ranking of the Mallows model. Let $\Phi_\sigma^{-1}(r) = (j_1, \ldots, j_m)$ be the insertion ranks. If we multiply the factors ϕ^{i-j_i} across $i \leq m$ this gives $\phi^{\sum_{i=1}^m i-j_i} = \phi^{d(r,\sigma)}$ by Proposition 6. This term $\phi^{d(r,\sigma)}$ is exactly the proportional probability of r in Mallows. The denominator of $\prod_{i=1}^m p_{ij_i}$ is $(1 + \phi)(1 + \phi + \phi^2) \cdots (1 + \phi + \cdots + \phi^{m-1})$ regardless of r—this is exactly the normalizing constant in Mallows model. Interestingly, this gives an alternate proof of the normalization constant in the Mallows model. □

Thus RIM offers a simple, useful way to sample rankings from the Mallows model while maintaining consistent partial rankings at each stage. In contrast to the rejection sampling approach of Condorcet/Mallows, RIM can be much more effective since it does not require the rejection of intransitive triads (which may occur with high probability if ϕ is large). We summarize the RIM approach from Mallows model:

RIM Sampling of Mallows

1. Let $\sigma = \sigma_1 \cdots \sigma_m$ be the reference ranking and ϕ the dispersion.
2. Start with an empty ranking r.
3. For $i = 1..m$:
   - Insert σ_i into r at rank position $j \leq i$ with probability $\phi^{i-j}/(1 + \phi + \cdots + \phi^{i-1})$.

RIM has worst-case quadratic running time (required number of draws from a Bernoulli distribution) when sampling from a Mallows model (this can be explained in much the same way as the complexity of insertion sort). However, the average-case time complexity can be much smaller, since insertions at each stage of the algorithm are likely to occur near the bottom of the partial ranking.

Proposition 7. *The expected time complexity of repeated insertion sampling for a Mallows model (σ, ϕ) is*

$$O\left(\min\left\{\frac{m(1 + \phi^{m+1})}{1 - \phi} - \frac{\phi(1 - \phi^m)}{(1 - \phi)^2}, m^2\right\}\right).$$

Proof. Suppose we have $O(1)$ access to biased coin flips. The implementation will be as follows. Place σ_1 in the first rank. Then loop for $i = 2$ to m. Let $p_{ij} = \phi^{i-j} / \sum_{j'=0}^{i-1} \phi^{j'}$. Sample a rank position j to insert σ_i: start with $j = i$, flip a coin with probability p_{ij}, if success insert at rank j. Otherwise decrease j by 1, flip a coin with probability $p_{ij}/(1 - \sum_{j'>j} p_{ij'})$, if success, insert at rank j, otherwise decrease j by 1 and repeat this process until $j = 1$. By the chain rule, the probability of insertion at rank j is exactly what Mallows model requires. For each σ_i, when the sampled insertion rank position is j, it would require at most $i - j + 1$ coin flips. The expected running time, i.e., total number of coin flips, if $\phi < 1$, is proportional to

$$\begin{aligned}\sum_{i=1}^{m} \frac{\sum_{j=0}^{i-1}(j+1)\phi^j}{\sum_{j=0}^{i-1}\phi^j} &= \sum_{i=1}^{m} \frac{1}{1-\phi} - i\phi^i \\ &\leq \frac{m(1+\phi^{m+1})}{1-\phi} - \frac{\phi(1-\phi^m)}{(1-\phi)^2}.\end{aligned}$$

This means one can effectively sample in linear time if ϕ is not too close to 1. If $\phi = 1$, the expected running time is $O(m^2)$. □

5.2.2.4 Sampling with Weighted Kendall-tau

To illustrate the flexibility of RIM, we show it can be used to sample from a Mallows model using a weighted Kendall-tau distance. For two rankings r and σ and insertion vector $\mathbf{j} = (j_1, \ldots, j_m)$ such that $\Phi_\sigma(\mathbf{j}) = r$, one can define a *weighted Kendall-tau distance* [111] with respect to positive weights $\mathbf{w} = (w_1, \ldots, w_m)$ as follows

$$d_{\mathbf{w}}(r, \sigma) = \sum_{i=1}^{m} w_i(i - j_i).$$

Recall that by Proposition 6, if $\mathbf{w} = \mathbf{1}$ (the all-1 vector), then $d_{\mathbf{w}}$ is the standard Kendall-tau distance. Otherwise, this weighted Kendall-tau is sensitive to the pairwise misorderings of top-ranked alternatives in σ.

One can sample from a Mallows model defined by $P_{\mathbf{w}}(r) \propto \exp(-d_{\mathbf{w}}(r, \sigma))$ using RIM as follows. Let $\phi_i = \exp(-w_i)$ for $i \leq m$. If we define the insertion probability of σ_i at position $j_i \leq i$ to be

$$\frac{\phi_i^{i-j_i}}{(1 + \phi_i + \cdots + \phi_i^{i-1})},$$

then the probability of generating r is proportional to

$$\exp\left(\sum_{i=1}^{m}(i-j_i)\ln\phi_i\right) = \exp\left(-d_{\mathbf{w}}(r,\sigma)\right).$$

5.2.3 A Mallows Mixture Model for Incomplete Preferences

While distributions such as Mallows or its mixture formulation [95] give rise to complete rankings, there is relatively little work on generative models for partial rankings, and in particular, models that generate arbitrary (consistent) sets of pairwise comparisons. We introduce such a generative model in this section upon which to base our subsequent learning and inference procedures given such pairwise evidence.

A *Mallows mixture distribution* with K components is parameterized by mixing proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, reference rankings $\boldsymbol{\sigma} = (\sigma^{(1)}, \ldots, \sigma^{(K)})$, and dispersion parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_K)$. Rankings are generated randomly by selecting one of the K components according to the multinomial distribution with parameters $\boldsymbol{\pi}$. We sometimes represent this with a unit *component indicator* vector $\mathbf{z} = (z_1, \ldots, z_K) \in \{0,1\}^K$ in which the only entry of $\mathbf{z}$ set to 1 is that of the selected component. If $z_k = 1$, then ranking r is drawn from the Mallows distribution with parameters $\sigma^{(k)}, \phi_k$.

In our model for partial preferences, we assume that each agent ℓ possesses a latent ranking r, where r is drawn from a mixture of Mallows distributions. We obtain the set of pairwise comparisons for ℓ by assuming a single additional parameter α which generates random pairs of alternatives. Intuitively, this reflects a process in which, given ℓ's latent ranking r, each pair of alternatives is selected independently with probability α, and ℓ's preference for that pair, as dictated by r, is revealed. That is,

$$P(v \mid r, \alpha) = \begin{cases} \alpha^{|v|}(1-\alpha)^{\binom{m}{2}-|v|} & \text{if } r \in \mathsf{C}(v), \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

This model reflects the relatively straightforward *missing at random assumption* [53], in which there is no correlation among those pairwise preferences that are missing/observed, nor any between observed pairs and the underlying ranking (e.g., the positions of the observed pairs). The missing at random assumption is not always realistic [89]. We also note that this model assumes a single global parameter α that indicates the expected degree of completeness of each agent ℓ's partial preferences. Allowing agent-specific completeness parameters α_ℓ and moving beyond "missing at random" are important directions. However, this model serves as a reasonable starting point for investigation. Figure 5.1

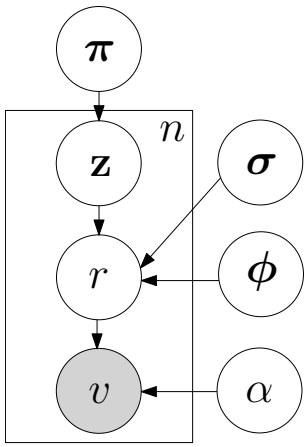

Figure 5.1: The generative model of partial preferences. Observed data v, a set of pairwise comparisons, is shaded.

illustrates a graphical model for the entire process. The resulting joint distribution is:

$$P(v, r, \mathbf{z} \mid \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}, \alpha) = P(v \mid r, \alpha) P(r \mid \mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\phi}) P(\mathbf{z} \mid \boldsymbol{\pi}). \tag{5.10}$$

In our basic inference and learning problem, we take the observed data to be a preference profile $\mathbf{v} = (v_1, \ldots, v_n)$ of n agents, and we let $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ denote the corresponding latent component memberships (i.e., $\mathbf{z}_i$ indicates the mixture component from which v_i is generated).

5.3 Related Work

There is a large literature on ranking in the machine learning, statistics, economics, and theory of computation communities. It includes a variety of approaches, evaluation criteria, heuristics and applications, driven by several distinct motivations. In this section we briefly review two somewhat distinct lines of research.

The first body of work is that on *rank aggregation*. Roughly speaking, the aim is to find the best *objective ranking* given complete or partial observations generated by some noisy process involving the (latent) objective ranking. For example, such a ranking may be a ranking of web pages expressing a typical user's (relative) degree of satisfaction with the pages. Observed information may consist of feedback, in the form of expert ratings or user preferences expressed implicitly via web page clicks on a search results page. In other applications, observed data may include partial rankings (e.g., in political elections), or pairwise comparisons (e.g., in sports leagues). Given such feedback, the ranking system will aggregate and optimize some objective function that attempts to capture user or

population satisfaction such as NDCG—common in the IR field—[22, 119], misordered pairs [29, 50, 68, 60], binary relevance [1, 108], and objectives from social choice theory (e.g., Kemeny, Borda rankings). For example, in machine learning, the area of *learning to rank (LETOR)* has been a topic of much research since the late 1990s, starting with the work of Cohen et al. [29]. Research into ranking systems often seeks strong *generalization* capabilities, in the sense that it can produce an objective ranking given a previously unencountered ranking problem using new attributes (e.g., rank web pages given a new search query). Much of this research has indeed been focused on web ranking applications (e.g., the Yahoo! Learning To Rank Challenge [21]). Recently Busa-Fekete et al. [23] have developed active learning algorithms for inferring certain distributional properties of the Mallows model.

There are also communities in statistics and computational social choice that are concerned with estimating the maximum likelihood ranking under some distributional assumptions. Often such models—for example, the Mallows and Plackett-Luce models discussed above—assume a central, modal or reference objective ranking at which the distribution is peaked. A fundamental problem is estimation of this objective ranking from a collection of ordinal preference data. For example, the Kemeny ranking can be interpreted as a maximum likelihood estimate of the modal ranking in a Mallows model [126]. Other such interpretations of common rank aggregation rules also exist [33, 37].

The above perspective, that of computing an *objective* ranking, applies to many situations (e.g., one would expect the ranking of web pages for a search query in "norovirus symptoms" to be objectively stable, since users will largely agree the informativeness of retrieved web pages). However, in many settings this is entirely inappropriate. When a group of individuals plans an activity together, such as going to a restaurant for dinner, the ranking of restaurants should clearly depend on the personal tastes and preferences of the individuals involved. In such cases, a distribution over a population's *subjective preferences* better reflects reality. A second, growing, body of work aims to assess (individual or aggregate/group) rankings of options, or decisions, by explicitly using, modelling or reasoning about the diversity of user preferences. This is a more general problem than that of objective rank aggregation. For example, the Netflix collaborative filtering competition has initiated much research on predicting a user's movie ratings given the ratings for other movies, including their own and those of other users. Other relevant research on such ranking work includes *label ranking* [64], which seeks to aggregate sparse preference data of "similar users" into personalized preferences.

In recent years there has been growing interest in applying probabilistic models of preferences from statistics, psychometrics, and econometrics to model a population's

preferences. This is the context in which our work is situated. We focus on learning such preference distributions, including multimodal distributions over preferences where each mode (cluster) corresponds to a "sub-type" within the population. Much recent research has focused on using the single-peaked Mallows model as a basis for multimodal mixture distributions. One of the first papers to propose an algorithm for learning Mallows mixtures is that of Murphy and Martin [95]. Their method assumes that training data takes the form of *complete* preference rankings (individual preferences), and has a running time that is factorial in the number of alternatives. Busse et al. [24] develop a tractable EM algorithm for Mallows mixtures where preferences are restricted to be of the top-t type. A recent extension by Meila and Chen [90] of Mallows mixtures allows for a Bayesian treatment in choosing the number of components using Dirichlet process mixtures, and offers experiments on considerably larger datasets. Recent work has also studied fitting temporal mixture models (a variation on the Bradley-Terry model) using EM [49].

Aside from mixture models, Lebanon and Mao [79] propose a non-parametric kernel density estimator for rankings, which places a "smooth Mallows bump" on each training preference. They derive an efficiently computable, closed-form formula for the evaluation of the estimator. However, they restrict their training data to partitioned preferences (see above), a more general concept than top-t rankings, but significantly less expressive than arbitrary pairwise comparisons. In contrast to our work, they do not address how to learn the kernel bandwidth parameter (see Section 5.7 for further discussion). There has been recent work on sampling algorithms for rankings that shares some similarities with the GRIM algorithm we develop here. This includes a sampling algorithm based on a generalization of the Plackett-Luce model [120], inspired by bipartite matching problems that occur in certain application domains. Biernacki and Jacques [9] propose a noisy insertion-sort model of rankings and develop EM algorithms for estimating its parameters. This is related to RIM but with some minor differences. However, none of this work addresses the question of sampling from a posterior distribution given partial preferences as evidence.

Apart from the Mallows model, the Plackett-Luce model has also been popular as a representation of preferences. Recent work on learning and inference with this model includes: an approach to Bayesian inference of the modal ranking [58], but where training preferences are limited to ranking of all of alternatives in some subset of alternatives; and a method for learning a mixture model given top-k preferences [56], with application to political voting data.

Huang and Guestrin [61] develop the *riffle independence model*, which partitions a

set of alternatives into two sets: a ranking of each set is generated stochastically (and independently); then a stochastic process is used to interleave or "riffle" the two resulting rankings to produce a combined ranking. The model can applied hierarchically, with the same process used to generate the required subrankings. Huang et al. [63] show that inference in this model is tractable for certain classes of observations. Of particular note is that fact that conditioning on partitioned preferences (which they term "partial ranking observations") can be accomplished efficiently.

5.4 Generalized Repeated Insertion Model

Our ultimate goal is to support effective learning and inference with Mallows models (and by extension, Mallows mixtures) given observed data or evidence in the form of partial preference profiles consisting of arbitrary pairwise comparisons. Sampling is, of course, an important aspect of this. The rejection sampling models discussed above can obviously be extended to accommodate pairwise observations, but are likely to be extremely inefficient. By contrast, while RIM provides a powerful tool for sampling from Mallows models (and mixtures), it samples unconditionally, without allowing for (direct) conditioning on evidence. In this section, we describe and analyze a generalized version of the RIM technique that permits conditioning at each insertion step. In fact, our *generalized repeated insertion model (GRIM)* can be used to sample from *arbitrary* rank distributions. We begin in Section 5.4.1 by describing GRIM in this general, abstract fashion. The primary focus of our theoretical and computational analysis will focus on its use for Mallows distributions.

5.4.1 Sampling from Arbitrary Ranking Distributions

We first present the *generalized repeated insertion model (GRIM)* abstractly as a means of sampling from *any* distribution over rankings. GRIM is based on a relatively simple insight, namely, that the chain rule allows us to represent any distribution over rankings in a concise way, as long as we admit *dependencies* in our insertion probabilities. Specifically, we allow the insertion probabilities for any alternative σ_i in the reference ranking to be *conditioned on the ordering* of the previously inserted alternatives $(\sigma_1, \ldots, \sigma_{i-1})$.

Let Q be any distribution over rankings and σ an (arbitrary) reference ranking. Recall that we can (uniquely) represent any ranking $r \in \mathsf{C}$ using σ and an insertion vector $\mathbf{j}^r = (j_1^r, \ldots, j_m^r) \in I$, where $r = \Phi_\sigma(\mathbf{j}^r)$. Thus Q can be represented by a distribution Q' over the space I of insertion vectors, i.e., $Q'(\mathbf{j}^r) = Q(r)$. Similarly, for $k < m$, any partial

ranking $r[k] = (r_1, \ldots, r_k)$ of the alternatives $S = \{\sigma_1, \ldots, \sigma_k\}$, can be represented by a partial insertion vector $\mathbf{j}[k] = (j_1^r, \ldots, j_k^r)$. Letting

$$Q(r[k]) = \sum_{r' \in \mathsf{C}} Q(r') \cdot \mathbf{1}[r_1 \succ r_2 \succ \cdots \succ r_k = r'|_S]$$

and

$$Q'(\mathbf{j}[k]) = \sum_{\mathbf{j}' \in I} Q'(\mathbf{j}') \cdot \mathbf{1}[\mathbf{j}'[k] = \mathbf{j}[k]],$$

we have $Q'(\mathbf{j}[k]) = Q(r[k])$. We define *conditional insertion probabilities*

$$p_{ij\,|\,\mathbf{j}[i-1]} = Q'(j_i = j \,|\, \mathbf{j}[i-1]). \tag{5.11}$$

This denotes the probability with which the ith alternative σ_i in the reference ranking is inserted at position $j \leq i$, *conditioned on the specific insertions* $(j_1^r, \ldots, j_{i-1}^r)$ of all previous alternatives. By the chain rule, we have

$$Q'(\mathbf{j}) = Q'(j_m|\mathbf{j}[m-1])Q'(j_{m-1}|\mathbf{j}[m-2]) \cdots Q'(\mathbf{j}[1]).$$

Suppose we apply RIM with conditional insertion probabilities $p_{ij|\mathbf{j}[i-1]}$ defined above; that is, we draw random insertion vectors $\mathbf{j}$ by sampling j_1 through j_m, in turn, but with each conditioned on the previously sampled components. The chain rule ensures that the resulting insertion vector is sampled from the distribution Q'. Hence the induced distribution over rankings $r = \Phi_\sigma(\mathbf{j})$ is Q. We call the aforementioned procedure the *generalized repeated insertion model (GRIM)*. Based on the arguments above, we have:

Theorem 12. *Let Q be any ranking distribution and σ a reference ranking. For any $r \in \mathsf{C}$, with insertion vector $\mathbf{j}^r$ (i.e., $r = \Phi_\sigma(\mathbf{j}^r)$), GRIM, using the insertion probabilities in Eq. 5.11, generates insertion vector $\mathbf{j}^r$ with probability $Q'(\mathbf{j}^r) = Q(r)$.*

For instance, GRIM can be used to sample from a (conditional) Mallows model given evidence in the form of pairwise comparisons, as shown in the following example.

Example 3. We illustrate GRIM using a simple example, sampling from a (conditional) Mallows model over $A = \{a, b, c\}$, with dispersion ϕ, given evidence $v = \{a \succ c\}$. Table 5.1 shows describes the steps in the process.

The resulting ranking distribution Q is given by the product of the conditional insertion probabilities: $Q(abc) = 1/(1+\phi)^2$; $Q(acb) = \phi/(1+\phi)^2$; and $Q(bac) = \phi/(1+\phi)$. As required, $Q(r) = 0$ if and only if r is inconsistent with evidence v.

Insert a, b		Insert c given ab		Insert c given ba	
r	Insertion Prob.	r	Insertion Prob.	r	Insertion Prob.
a	$P(j_a=1)=1$	cab	$P(j_c=1)=0$	cba	$P(j_c=1)=0$
ab	$P(j_b=1)=\frac{1}{1+\phi}$	acb	$P(j_c=2)=\frac{\phi}{1+\phi}$	bca	$P(j_c=2)=0$
ba	$P(j_b=2)=\frac{\phi}{1+\phi}$	abc	$P(j_c=3)=\frac{1}{1+\phi}$	bac	$P(j_c=3)=1$

Table 5.1: Example of using GRIM to sample from a conditional Mallows model.

5.4.2 Sampling from Mallows Posteriors

We now develop and analyze several techniques for sampling from (mixtures of) Mallows models given partial preference profiles as evidence. We use the term *Mallows posterior* to refer to the conditional distribution that arises from incorporating evidence—in the form of a set of pairwise comparisons—into a known Mallows model. This is the primary inference task facing a system making predictions about a specific user's preferences given pairwise evidence from that user, assuming a reasonably stable population model. This stands in contrast to the more general problem of learning the parameters of a Mallows model (a problem we address in Section 5.5).

5.4.2.1 Intractability of Sampling

One key difficulty with enabling inference conditioned on pairwise comparisons is the intractability of the posterior. In the above model (Eq. 5.10), where agent ℓ's incomplete preference v_ℓ is observed, it is intractable to work with the posterior $P(r, \mathbf{z}|v_\ell, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}, \alpha)$ even when the mixture model has a single component, a fact we prove below. One typical approach is to rely on sampling to estimate the posterior. To this end, we develop a polynomial-time posterior sampling algorithm based on GRIM, but relying on approximation of the relevant conditional insertion probabilities.

While GRIM allows sampling from arbitrary distributions over rankings, as presented above it is largely a theoretical device, since it requires inference to compute the required conditional probabilities. Thus to use GRIM to sample from a Mallows posterior, given arbitrary pairwise comparisons v, we must first derive these required terms. The Mallows posterior is given by:

$$P_v(r) = P(r \mid v) = \frac{\phi^{d(r,\sigma)}}{\sum_{r' \in \mathsf{C}(v)} \phi^{d(r',\sigma)}} \cdot \mathbf{1}[r \in \mathsf{C}(v)], \tag{5.12}$$

which requires summing over an intractable number of rankings to compute the normalization constant.

We could use RIM for rejection sampling: sample unconditional insertion ranks, and reject a ranking at any stage if it is inconsistent with v. However, this is impractical because of the high probability of rejection. One can also modify the pairwise comparison sampling model (see Section 5.2.2.2) to reject inconsistent pairwise comparisons. However, if $|v|$ is small relative to m, then for values of ϕ that are not too small, the probability of rejection is very high. For instance, if ϕ is close to 1, $m = 120$ and 30 alternatives appear in v, any three alternatives the probability of a cyclic triad for any triple (e.g., $a \succ b, b \succ c, c \succ a$) is $\approx 1/4$. The 90 alternatives unconstrained by v can be divided into 30 groups of 3 alternatives, hence the probability that a cycle occurs among at least one triad is at least $1 - (3/4)^{30} \approx 0.9998$. This is a lower bound on the probability of rejection, showing rejection sampling to be impractical in many settings.

The main obstacle to using GRIM for sampling is computation of the insertion probabilities of a specific alternative given the inserted positions all previous alternatives, as given by Eq. 5.11, when Q' (more precisely, the corresponding Q) is the Mallows posterior. This essentially involves computing a high-order marginal over rankings, and turns out to be #P-hard, even with a uniform distribution over $\mathsf{C}(v)$. The following result on the complexity of counting linear extensions of a partial order will be useful below:

Theorem 13 (Brightwell and Winkler [20])**.** *Given a partial order v, computing the number of linear extensions of v, that is $|\mathsf{C}(v)|$, is #P-complete.*

To show that computing a function $f(x)$ is #P-hard for input x, it is sufficient to show that a #P-complete problem can be reduced to it in polynomial time.

Proposition 8. *Given v, a reference ordering σ, a partial ranking $r_1 \cdots r_{i-1}$ over $\sigma_1, \ldots, \sigma_{i-1}$, and $j \leq i$, computing the probability of inserting σ_i at rank j with respect to the uniform Mallows posterior P (i.e., computing $P(r) \propto \mathbf{1}[r \in \mathsf{C}(v)]$) is #P-hard.*

Proof. We reduce the problem of counting the number of linear extensions of incomplete preferences v, which is a #P-complete problem, to that of computing the desired insertion probabilities, showing the problem to be #P-hard. Given v, notice that any $r = r_1 \ldots r_m \in \mathsf{C}(v)$ has a uniform posterior probability of $1/|\mathsf{C}(v)|$. Let $\Phi_\sigma^{-1}(r) = (j_1, \ldots, j_m)$. Assume the existence of an algorithm f to compute the required insertion probabilities. We can use it to solve the counting problem as follows: we use f to compute $p_{ij_i} = \Pr(\text{insert } \sigma_i \text{ at rank } j_i \mid r|_{\{\sigma_1,\ldots,\sigma_{i-1}\}})$ with partial order v for each $i \in \{2, \ldots, m\}$ (i.e., $m - 1$ applications of f). By Theorem 12, we know the posterior probability of r is $1/|\mathsf{C}(v)| = \prod_i p_{ij_i}$; thus we can compute $|\mathsf{C}(v)|$ by inverting the product of the insertion probabilities. Note that this reduction can be computed in polynomial

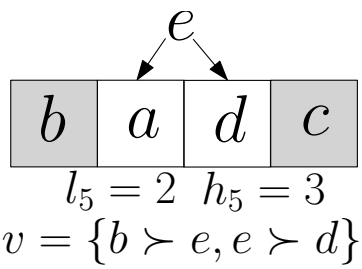

Figure 5.2: Valid insertion ranks for e are $\{l_5, \ldots, h_5\} = \{2, 3\}$ given previous insertions and constraints v.

time: we can construct any $r \in \mathsf{C}(v)$ by using a topological sort algorithm, and we require only $m - 1$ calls to the algorithm insertion algorithm f. □

This suggests it is hard to sample exactly, and that computing the normalization constant in a Mallows posterior is difficult. This would also imply a computational complexity obstacle in the work on non-parametric estimators with a Mallows kernel [79] for an *arbitrary* set of pairwise comparisons. Nevertheless we develop an *approximate sampler* AMP that is computationally very efficient. While its approximation quality can be quite poor in the worst-case, we see below that, empirically, it produces excellent posterior approximations. We also derive bounds that delineate circumstances under which it will provide approximations with low error.

5.4.2.2 AMP: An Approximate Sampler

AMP is based on the same intuitions as those illustrated in Example 3, where instead of computing the correct insertion probabilities, we use the (unconditional) insertion probabilities used by RIM, but subject to constraints imposed by v. First, we compute the transitive closure $\mathsf{tc}(v)$ of v. Then we use a modified repeated insertion procedure where at each step, the alternative being inserted can only be placed in positions that do not contradict $\mathsf{tc}(v)$. We can show that the valid insertion positions for any alternative, given v, form a *contiguous region* of the ranking (see Figure 5.2 for an illustration).

Proposition 9. *Given partial preference v, let the insertion of $i-1$ alternatives $\sigma_1, \ldots, \sigma_{i-1}$ induce a ranking $r_1 \cdots r_{i-1}$ that is consistent with $\mathsf{tc}(v)$. Let $L_i = \{i' < i | r_{i'} \succ_{\mathsf{tc}(v)} \sigma_i\}$ and $H_i = \{i' < i | r_{i'} \prec_{\mathsf{tc}(v)} \sigma_i\}$. Then inserting σ_i at rank j is consistent with $\mathsf{tc}(v)$ if and*

Algorithm 3 AMP Approximate Mallows Posterior

Input: v, σ, ϕ

1: $r \leftarrow \sigma_1$
2: **for** $i = 2..m$ **do**
3: Calculate l_i and h_i from Eq. 5.13 and 5.14.
4: Insert σ_i in r at rank $j \in \{l_i, \ldots, h_i\}$ with probability $\frac{\phi^{i-j}}{\sum_{l_i \leq j' \leq h_i} \phi^{i-j'}}$.
5: **end for**

Output: r

only if $l_i \leq j \leq h_i$, *where*

$$l_i = \begin{cases} 1 & \text{if } L_i = \emptyset \\ \max(i' \in L_i) + 1 & \text{otherwise,} \end{cases} \tag{5.13}$$

$$h_i = \begin{cases} i & \text{if } H_i = \emptyset \\ \min(i' \in H_i) & \text{otherwise.} \end{cases} \tag{5.14}$$

Proof. Inserting σ_i at any rank position less than l_i is impossible since either $l_i = 1$ (we can't insert in rank 0) or σ_i lies above r_{l_i}, which contradicts the requirement imposed by $\mathsf{tc}(v)$ that r_{l_i} must be ranked higher. A similar argument can be made for inserting in rank below h_i since r_{h_i} needs to be below σ_i. Finally, inserting into any rank in $\{l_i, \ldots, h_i\}$ does not violate $\mathsf{tc}(v)$ since the alternative will be inserted below *all* alternatives that must precede it in $\mathsf{tc}(v)$ and *all* alternatives that must succeed it. □

Proposition 9 immediately suggests an implementation of the GRIM algorithm, AMP, for approximate sampling of the Mallows posterior—AMP is outlined in Alg. 3. It first initializes ranking r with σ_1 at rank 1. Then for each $i = 2 \ldots m$, it computes l_i, h_i and inserts σ_i at rank $j \in \{l_i, \ldots, h_i\}$ with probability proportional to ϕ^{i-j}. Note that $\mathsf{tc}(v)$, which is required as part of the algorithm, can be computed via a modified depth-first search. AMP induces a sampling distribution $\hat{P}_v$ that does not match the posterior P_v exactly: indeed the KL-divergence between the two can be severe, as the following example shows.

Example 4. Let $A = \{a_1, \ldots a_m\}$ and $v = a_2 \succ a_3 \succ \cdots \succ a_m$. Let P be the uniform Mallows prior ($\phi = 1$) with $\sigma = a_1 \cdots a_m$. There are m rankings in $\mathsf{C}(v)$, one ranking r_i for each placement of a_1 into rank position $1 \leq i \leq m$. That is, $r_1 = a_1 a_2 \cdots a_m$ and $r_i = a_2 \cdots a_i a_1 a_{i+1} \cdots a_m$ for $i \geq 2$. The true Mallows posterior P_v is uniform over $\mathsf{C}(v)$. But AMP induces an approximation with $\hat{P}_v(r_i) = 2^{-i}$ for $i \leq m-1$ and $\hat{P}_v(r_m) = 2^{-m-1}$. To see this, note that to construct r_i, AMP would need to insert alternatives $a_2, \ldots, a_i$

successively, each with probability 1/2, above a_1. Then a_{i+1} must be inserted below a_1 with probability 1/2, and finally the remaining alternatives $a_{i+2}, \ldots, a_m$ can only be inserted at the bottom (with probability 1). Hence, the KL-divergence between P_v and $\hat{P}_v$ is

$$\begin{aligned}
\mathrm{KL}(P_v||\hat{P}_v) &= \sum_{i=1}^{m} P_v(r_i) \log_2 \left(\frac{P_v(r_i)}{\hat{P}_v(r_i)} \right) \\
&= \left[\sum_{i=1}^{m-1} \frac{1}{m} \log_2 \frac{1/m}{2^{-i}} \right] + \frac{1}{m} \log_2 \frac{1/m}{2^{-m+1}} \\
&= 1 - \frac{1}{m} + \frac{m-1}{2} - \log_2 m \, .
\end{aligned}$$

5.4.2.3 Statistical Properties of AMP

Example 4 shows that AMP may provide poor approximations in the worst-case; however we will see below (Section 5.6) that it performs very well in practice. We can also prove interesting properties, and provide theoretical guarantees of exact sampling in important special cases.

We first observe that AMP always produces a valid ranking; in other words, valid insertion positions always exist given any consistent v.

Proposition 10. *For all $i \geq 2$ and all rankings of alternatives $\sigma_1, \ldots, \sigma_{i-1}$ that is consistent with v, we have that $l_i \leq h_i$, where l_i and h_i are defined in Eq. 5.13 and 5.14, respectively. That is,* AMP *always has a position at which to insert alternative σ_i.*

Proof. Let r be a ranking of $\sigma_1, \ldots, \sigma_{i-1}$ consistent with v. Let x be the lowest ranking alternative in r such that $x \succ_{\mathsf{tc}(v)} \sigma_i$ and y the highest-ranked alternative in r with $y \prec_{\mathsf{tc}(v)} \sigma_i$. By transitivity, $x \succ_{\mathsf{tc}(v)} y$. Now if $h_i < l_i$ (as defined in terms of r) this implies $y \succ_r x$, but this contradicts the assumption that r is consistent with v. □

Furthermore, the approximate posterior has the same support as the true posterior:

Proposition 11. *The support of the distribution over rankings as defined by* AMP *is equal to $\mathsf{C}(v)$ which is equal to the support of the Mallows posterior as given in Eq. 5.12.*

Proof. By Proposition 9, the algorithm never violates the constraints in $\mathsf{tc}(v)$, and by Proposition 10, it will always have at least one valid insertion position. Hence the algorithm always outputs a ranking consistent with v. Now, let $r \in \mathsf{C}(v)$ and $\Phi_\sigma^{-1}(r) = (j_1, \ldots, j_m)$ be its corresponding insertion vector. We show that for all $i \leq m$, $j_i \in \{l_i, \ldots, h_i\}$. If this is not true, then there exists a smallest $i' \leq m$ such that

$j_{i'} \notin \{l_{i'}, \ldots, h_{i'}\}$ (note $i' \geq 2$ since the first alternative is always inserted at the first position). However, Proposition 9 asserts that this insertion rank would lead to a ranking inconsistent with v—so this is not possible. Since AMP places positive probability on any insertion position in $\{l_i, \ldots, h_i\}$ then r has positive probability under AMP. □

Proposition 12. *For any $r \in \mathsf{C}(v)$, the probability* AMP *will output r is*

$$\hat{P}_v(r) = \frac{\phi^{d(r,\sigma)}}{\prod_{i=1}^{m}(\phi^{i-h_i} + \phi^{i-h_i+1} + \cdots + \phi^{i-l_i})}. \tag{5.15}$$

Proof. Let $\Phi_\sigma^{-1}(r) = (j_1, \ldots, j_m)$ be the insertion ranks. We have already established in Proposition 11 that AMP puts positive probability on these valid insertion ranks. In fact the probability of r under the algorithm (see Alg. 3) is

$$\begin{aligned}\prod_{i=1}^{m} \frac{\phi^{i-j_i}}{(\phi^{i-l_i} + \phi^{i-l_i-1} + \cdots + \phi^{i-h_i})} &= \frac{\phi^{\sum_{i=1}^{m} i-j_i}}{\prod_{i=1}^{m}(\phi^{i-l_i} + \phi^{i-l_i-1} + \cdots + \phi^{i-h_i})} \\ &= \frac{\phi^{d(r,\sigma)}}{\prod_{i=1}^{m}(\phi^{i-l_i} + \phi^{i-l_i-1} + \cdots + \phi^{i-h_i})},\end{aligned}$$

where the last equality comes from Proposition 6. □

Using this result we can show that if v lies in the class of partitioned preferences, AMP's induced distribution is exactly the Mallows posterior:

Proposition 13 (Lebanon and Mao [79])**.** *Let σ be a reference ranking. Let v be a partitioned preference (see Definition 18) with partition $A_1, \ldots, A_q$ of A. Let $\delta = |\{(x, y) | y \succ_\sigma x, x \in A_i, y \in A_j, i, j \in [q], i < j\}|$, which is the number of pairs of alternatives, that span different subsets of the partition, that are misordered with respect to σ. Then*

$$\delta = \sum_{i=1}^{q-1} \sum_{x \in A_i} \sum_{j=i+1}^{q} \sum_{y \in A_j} \mathbf{1}[y \succ_\sigma x], \tag{5.16}$$

$$\sum_{r \in \mathsf{C}(v)} \phi^{d(r,\sigma)} = \phi^\delta \prod_{i=1}^{q} \prod_{j=1}^{|A_i|} (1 + \phi + \phi^2 + \cdots + \phi^{j-1}). \tag{5.17}$$

Notice that Eq. 5.17 represents the normalization constant in Mallows posterior. The intuition underlying Eq. 5.17 is that, for any $r \in \mathsf{C}(v)$, the misorderings contributed by alternatives that span two subsets, as given by δ, are the same (hence the leading factor) whereas within a subset A_i alternatives can be ordered arbitrarily (hence the product of normalization constants for $|A_i|$).

Proposition 14. *Given a partitioned preference v, the distribution induced by* AMP*, $\hat{P}_v$, is equal to the true Mallows posterior P_v.*

Proof. Since the numerator in Eq. 5.15 (which denotes the probability that AMP outputs r) is the same as the proportional probability of the Mallows posterior, it is sufficient to show that the denominator in Eq. 5.15 equals the Mallows posterior normalization constant given by Eq. 5.17. Suppose $\sigma = \sigma_1 \cdots \sigma_m$. Let v be a partitioned preference $A_1, \ldots, A_q$. Consider alternatives in A_i such that $\sigma|_{A_i} = \sigma_{t_1}\sigma_{t_2}\cdots\sigma_{t_{|A_i|}}$ (i.e., the ranking of alternatives in A_i according to σ). For any $k \in \{1, \ldots, |A_i|\}$, suppose alternatives $A' = \{\sigma_1, \ldots, \sigma_{t_k-1}\}$ are inserted. The structure of the resulting ranking is as follows: the alternatives $(A_1 \cup A_2 \cup \cdots \cup A_{i-1}) \cap A'$ must lie at the top of the ranking; the alternatives $A_i \cap A' = \{\sigma_{t_1}, \ldots, \sigma_{t_{k-1}}\}$ are in the middle; and $B_{t_k} = (A_{i+1} \cup \cdots \cup A_q) \cap A'$ are at bottom. When inserting σ_{t_k} at rank j, we have $j \in \{l_{t_k}, \ldots, h_{t_k}\}$, where $h_{t_k} = t_k - |B_{t_k}|$ and $l_{t_k} = h_{t_k} - |A_i \cap A'| = t_k - (k-1) - |B_{t_k}|$. Hence σ_{t_k} is inserted at rank j with probability

$$\frac{\phi^{t_k - j}}{\phi^{t_k - h_{t_k}} + \cdots + \phi^{t_k - l_{t_k}}} = \frac{\phi^{t_k - j}}{\phi^{|B_{t_k}|} + \cdots \phi^{k-1+|B_{t_k}|}}.$$

The denominator can be written $\phi^{|B_{t_k}|}(1 + \cdots + \phi^{k-1})$. Observe that B_{t_k} consists of all alternatives from A' that are above σ_{t_k} in σ, but are below it in v (since all such alternatives belong to $A_{i+1} \cup \cdots \cup A_q$). So $\sum_{k=1}^{|A_i|} |B_{t_k}|$ is the total number of pairs (x, y), where $x \in A_i$ and $y \in A_{i+1} \cup \cdots \cup A_q$, that are misordered with respect to σ. Thus inserting alternatives in A_i contributes a factor of

$$\prod_{k=1}^{|A_i|} \phi^{|B_{t_k}|}(1 + \cdots + \phi^{k-1}) = \phi^{\sum_{x \in A_i} \sum_{j=i+1}^{q} \sum_{y \in A_j} \mathbf{1}[y \succ_\sigma x]} \prod_{k=1}^{|A_i|} (1 + \cdots + \phi^{k-1})$$

to the denominator in Eq. 5.15. Once all alternatives have been inserted, the denominator becomes:

$$\phi^{\sum_{i=1}^{q} \sum_{x \in A_i} \sum_{j=i+1}^{q} \sum_{y \in A_j} \mathbf{1}[y \succ_\sigma x]} \prod_{i=1}^{q} \prod_{k=1}^{|A_i|} (1 + \cdots + \phi^{k-1}).$$

This is exactly the Mallows posterior normalization constant in Eq. 5.17. □

As a consequence, AMP provides exact sampling in the case of partitioned preferences, In general, this is not the case with arbitrary partial preferences (pairwise comparisons). We now derive bounds on the relative error of AMP's posterior, bounding the ratio between the sample probability of an arbitrary ranking r for AMP and the true posterior probability. The main technical challenge is deriving a bound on the Mallows posterior normalization constant. We can obtain an upper bound by exploiting the pairwise

comparison interpretation of the Mallows model (see Section 5.2.2.2).

Theorem 14 (Upper Bound on Normalization Constant)**.** *Let σ be a reference ranking, $\phi \in (0, 1]$ and v a preference. The Mallows posterior normalization constant is upper bounded by*

$$\sum_{r \in \mathsf{C}(v)} \phi^{d(r,\sigma)} \leq \phi^{d(v,\sigma)}(1+\phi)^{\binom{m}{2}-d(v,\sigma)-s(v,\sigma)}. \tag{5.18}$$

Proof. The LHS of Eq. 5.18 can be written in terms Eq. 5.3, by setting $\phi = p/(1-p)$ (see Section 5.2.2.2 for derivations of the pairwise comparison interpretation of Mallows) as follows:

$$\begin{aligned} \sum_{r \in \mathsf{C}(v)} \phi^{d(r,\sigma)} &= Z \cdot \sum_{r \in \mathsf{C}(v)} P(r|\sigma, p) \\ &= Z \cdot \frac{1}{Z'} \sum_{r \in \mathsf{C}(v)} \prod_{\{x,y\} \subseteq A} \begin{cases} p & \text{if } r \text{ and } \sigma \text{ disagree on } x, y \\ 1-p & \text{otherwise,} \end{cases} \end{aligned} \tag{5.19}$$

where $p = \phi/(1+\phi)$, Z' is given by Eq. 5.5 and Z is given by Eq. 5.2, thus the constant in front simplifies to $1/(1-p)^{\binom{m}{2}}$. Since r must be consistent with v, if x and y are comparable under v, then r must be agree with v on (x, y), i.e., if $x \succ_{\mathsf{tc}(v)} y$ then $x \succ_r y$. So

$$P(r|\sigma, p) = \frac{1}{Z'} p^{d(v,\sigma)}(1-p)^{s(v,\sigma)} \prod_{\{x,y\} \notin \mathsf{tc}(v)} \begin{cases} p & \text{if } r \text{ and } \sigma \text{ disagree on } x, y \\ 1-p & \text{otherwise.} \end{cases}$$

Hence, since $\mathsf{C}(v)$ is contained in the set of all *intransitive* relations on A that is consistent with comparisons in $\mathsf{tc}(v)$, we must have (for $k = \binom{m}{2} - d(v,\sigma) - s(v,\sigma)$)

$$\begin{aligned} \sum_{r \in \mathsf{C}(v)} P(r|\sigma, p) &\leq \frac{1}{Z'} p^{d(v,\sigma)}(1-p)^{s(v,\sigma)} \sum_{z \in \{0,1\}^k} \prod_{i=1}^{k} p^{z_i}(1-p)^{1-z_i}, \\ &= \frac{1}{Z'} p^{d(v,\sigma)}(1-p)^{s(v,\sigma)}. \\ Z \cdot \sum_{r \in \mathsf{C}(v)} P(r|\sigma, p) &\leq \frac{1}{(1-p)^{\binom{m}{2}}} p^{d(v,\sigma)}(1-p)^{s(v,\sigma)}. \end{aligned} \tag{5.20}$$

Combining Eq. 5.20 with Eq. 5.19, and noting that $p = \phi/(1+\phi)$, we obtain Eq. 5.18. □

Eq. 5.18 tells us if $d(v, \sigma)$ increases (i.e., v increasingly disagrees with σ), then the first factor dominates and upper bound gets smaller—this reflects our natural intuitions

since the set $\mathsf{C}(v)$ gets "further away" from reference ranking σ and hence its probability mass is small. We also see that if $|\mathsf{tc}(v)|$ is small, then $d(v,\sigma) + s(v,\sigma)$ is small and the upper bound increases since the second factor dominates. This too makes sense because $\mathsf{C}(v)$ is large and has greater probability mass. If $s(v,\sigma)$ is large, more constraints are placed on v, hence $\Pr[\mathsf{C}(v)]$ is smaller, and likewise the upper bound decreases. The following example illustrates that this bound may be quite loose in some cases, but tight in others.

Example 5. Consider again the partial ranking evidence from Example 4, where $v = a_2 \succ \cdots \succ a_m$, the alternatives are $\{a_1, \ldots, a_m\}$, and our reference ranking is $\sigma = a_1 a_2 \cdots a_m$. Recall that there are m rankings in $\mathsf{C}(v)$, one ranking r_i for each placement of a_1 into rank position i. Now the term on the LHS of Eq. 5.18, i.e., the true value of the normalization constant, is

$$\sum_{i=1}^{m} \phi^{d(r_i,\sigma)} = 1 + \phi + \phi^2 + \cdots + \phi^m.$$

Note that $d(v,\sigma) = 0$ and $s(v,\sigma) = \binom{m-1}{2}$ since all pairwise comparisons in $\mathsf{tc}(v)$ agree with σ. Thus, the term on the RHS of Eq. 5.18, i.e., the upper bound is

$$\phi^0 (1+\phi)^{\binom{m}{2} - 0 - \binom{m-1}{2}} = (1+\phi)^{m-1}.$$

This upper bound on the normalization constant gets tight as $\phi \to 0$, but becomes exponentially loose in m as $\phi \to 1$.

Before we derive a lower bound, we introduce some notions from order theory.

Definition 21. Let v be a partial preference. An *anti-chain* of v is a subset X of A such that for every $x, y \in X$ they are incomparable under $\mathsf{tc}(v)$. A *maximum anti-chain* is an anti-chain whose size is at least the size of any anti-chain. The *width* of v, $\mathsf{w}(v)$ is the size of a maximum anti-chain of v.

Theorem 15 (Lower Bound on Normalization Constant)**.** *Let σ be a reference ranking, and $\phi \in (0,1]$. Let X be a maximum anti-chain of v, $Y = \{a \in A \backslash X \mid \exists x \in X, a \succ_{\mathsf{tc}(v)} x\}$ and $Z = A \backslash (X \cup Y)$. Let $\delta = |\{(x,y) | x \in X, y \in Y, x \succ_\sigma y\}| + |\{(y,z) | y \in Y, z \in Z, z \succ_\sigma y\}| + |\{(x,z) | x \in X, z \in Z, z \succ_\sigma x\}|$. Denote by $\mathsf{tc}(v)|_Y$ and $\mathsf{tc}(v)|_Z$ the transitive closure of v restricted to the subsets Y and Z, respectively. Also let $\mathsf{C}(\mathsf{tc}(v)|_Y)$ denote those*

rankings over Y that are consistent with $\mathsf{tc}(v)|_Y$, and similarly for $\mathsf{C}(\mathsf{tc}(v)|_Z)$. We have,

$$\sum_{r\in\mathsf{C}(v)} \phi^{d(r,\sigma)} \geq \phi^{\delta} \left[\sum_{r\in\mathsf{C}(\mathsf{tc}(v)|_Y)} \phi^{d(r,\sigma|_Y)}\right] \left[\sum_{r\in\mathsf{C}(\mathsf{tc}(v)|_Z)} \phi^{d(r,\sigma|_Z)}\right] \prod_{i=1}^{\mathsf{w}(v)} \sum_{j=0}^{i-1} \phi^{j} \tag{5.21}$$

Proof. We first show that $Z' = \{a \in A\backslash X \mid \exists x \in X, x \succ_{\mathsf{tc}(v)} a\} = Z$. If $a \in A\backslash X$ does not belong to Y then it must be comparable to at least one element in $x \in X$ otherwise we can add it to Y and obtain a larger anti-chain. Hence, since a is not in Y, then $x \succ_{\mathsf{tc}(v)} a$. Also, note that if $a \in Y$ then $a \notin Z'$. This is because if a belonged to both Y and Z, then there exists $x_1, x_2 \in X$ such that $x_1 \succ_{\mathsf{tc}(v)} a$ and $a \succ_{\mathsf{tc}(v)} x_2$ this would mean $x_1 \succ_{\mathsf{tc}(v)} x_2$ which contradicts the anti-chain property of X. For a particular alternative in X, alternatives in Y are either incomparable to it or must be preferred to it, similarly alternatives in Z are either incomparable or must be dis-preferred to it.

This also implies no alternative in Z can be preferred over alternatives in Y since if this were to happen, i.e., if $z \succ_{\mathsf{tc}(v)} y$ where $z \in Z, y \in Y$, then $\exists x \in X$ such that $y \succ_{\mathsf{tc}(v)} x$, this implies $z \succ_{\mathsf{tc}(v)} x$ which is impossible from the above observation that $Z \cap Y = \emptyset$.

Consider all rankings $\widetilde{\mathsf{C}}(v)$ where we place alternatives of Y at the top, X in the middle and Z at the bottom. Within Y and Z we rank alternatives respecting $\mathsf{tc}(v)$ and since X is an anti-chain, rank these alternatives without restrictions. That is

$$\widetilde{\mathsf{C}}(v) = \{r | \forall y \in Y, x \in X, z \in Z, y \succ_r x, x \succ_r z, r|_Y \in \mathsf{C}(\mathsf{tc}(v)|_Y), r|_Z \in \mathsf{C}(\mathsf{tc}(v)|_Z)\}.$$

Now we argue $\widetilde{\mathsf{C}}(v) \subseteq \mathsf{C}(v)$. Note that we satisfy preference constraints when ranking within Y, X and Z. Also as we showed above, alternatives in Y are never dis-preferred to alternatives in X or Z and alternatives in X are never dis-preferred to alternatives in Z.

For the lower bound, first observe if $r \in \widetilde{\mathsf{C}}(v)$ then $d(r,\sigma) = d(r|_Y, \sigma|_Y) + d(r|_X, \sigma|_X) + d(r|_Z, \sigma|_Z) + \delta$ where δ is defined in the theorem as the number of misorderings of alternatives across X, Y, Z, which is independent of r. Hence,

$$\sum_{r\in\mathsf{C}(v)} \phi^{d(r,\sigma)} \geq \sum_{r\in\widetilde{\mathsf{C}}(v)} \phi^{d(r,\sigma)} = \phi^{\delta} \left[\sum_{r\in\mathsf{C}(\mathsf{tc}(v)|_Y)} \phi^{d(r,\sigma|_Y)}\right] \left[\sum_{r\in\mathsf{C}(\mathsf{tc}(v)|_X)} \phi^{d(r,\sigma|_X)}\right] \left[\sum_{r\in\mathsf{C}(\mathsf{tc}(v)|_Z)} \phi^{d(r,\sigma|_Z)}\right].$$

Finally, it can be seen that the sum inside the third factor is exactly the normalization constant of an unconstrained Mallows model with $|X| = \mathsf{w}(v)$ alternatives, and hence equal to $\prod_{i=1}^{\mathsf{w}(v)} \sum_{j=0}^{i-1} \phi^j$, the second and fourth factors involve sums over rankings of Y and Z consistent with $\mathsf{tc}(v)$. This proves the lower bound. □

While the lower bound is not presented in a convenient closed form, it is useful nonetheless if $\mathsf{w}(v)$ is large: if there are few preference constraints in v (e.g., v involves only a small subset of alternatives) we expect $\mathsf{C}(v)$ to be large and hence have higher probability mass. We recover the true Mallows normalization constant if $v = \emptyset$ since $\mathsf{w}(v) = m$. If v is highly constrained—$\mathsf{C}(v)$ has smaller probability mass—then $\mathsf{w}(v)$ is small, but so are the factors involving summations in Eq. 5.21. Note that ϕ^δ decreases as the number of comparisons in v that disagree with σ increases; this again corresponds to intuition.

With these bounds in hand, we can bound the quality of the posterior estimate $\hat{P}_v(r)$ produced by AMP:

Corollary 1. *Let L and U be the lower and upper bound as in Theorems 15 and 14, respectively. Then for $r \in \mathsf{C}(v)$, where l_i and h_i are defined in Proposition 9,*

$$\frac{L}{\prod_{i=1}^{m} \sum_{j=l_i}^{h_i} \phi^{i-j}} \leq \frac{\hat{P}_v(r)}{P_v(r)} \leq \frac{U}{\prod_{i=1}^{m} \sum_{j=l_i}^{h_i} \phi^{i-j}} \tag{5.22}$$

Proof. $\hat{P}_v(r)$ has the form given in Proposition 12 while $P_v(r) \propto \phi^{d(r,\sigma)}$. Then apply upper and lower bounds on the normalizing constant of $P_v(r)$. □

5.4.2.4 MMP: An MCMC Sampler Based on AMP

While AMP may have (theoretically) poor worst-case performance, we use it as the basis for a *statistically sound* sampler MMP, by exploiting AMP to propose new rankings for the Metropolis algorithm. With Eq. 5.15, we can derive the acceptance ratio for Metropolis. At step $t+1$ of Metropolis, let $r^{(t)}$ be the previous sampled ranking. Ranking r, proposed by AMP independently of $r^{(t)}$, will be accepted as the $t+1$st sample $r^{(t+1)}$ with probability

Algorithm 4 MMP Sample Mallows Posterior using Metropolis

Input: v, σ, ϕ, number of steps T

1: **for** $t = 1..T$ **do**

2: $r \leftarrow \mathsf{AMP}(v, \sigma, \phi)$

3: $a \sim$ Uniform[0,1]

4: $r^{(t)} \leftarrow \begin{cases} r & \text{if } t = 1 \text{ or } a \leq a^*(r, r^{(t-1)}) \\ r^{(t-1)} & \text{otherwise} \end{cases}$

5: **end for**

Output: $r^{(T)}$

$a^*\left(r, r^{(t)}\right)$, where:

$$\begin{aligned} a^*\left(r, r^{(t)}\right) &= \min\left(1, \frac{\phi^{d(r,\sigma)}/Z_v}{\phi^{d(r^{(t)},\sigma)}/Z_v} \frac{\frac{\phi^{d(r^{(t)},\sigma)}}{\prod_{i=1}^{m} \phi^{i-h_i^t} + \phi^{i-h_i^t+1} + \cdots + \phi^{i-l_i^t}}}{\frac{\phi^{d(r,\sigma)}}{\prod_{i=1}^{m} \phi^{i-h_i} + \phi^{i-h_i+1} + \cdots + \phi^{i-l_i}}}\right) \\ &= \min\left(1, \prod_{i=1}^{m} \begin{cases} \frac{h_i - l_i + 1}{h_i^t - l_i^t + 1} & \text{if } \phi = 1 \\ \frac{\phi^{h_i^t - h_i}(1 - \phi^{h_i - l_i + 1})}{1 - \phi^{h_i^t - l_i^t + 1}} & \text{otherwise} \end{cases}\right). \end{aligned} \tag{5.23}$$

Here the l_is and h_is are defined as in Eqs. 5.13 and 5.14, respectively (with respect to r; and l_i^t), and h_i^t are defined similarly, but with respect to $r^{(t)}$. The term $Z_v = \sum_{r' \in \mathsf{C}(v)} \phi^{d(r',\sigma)}$ is the normalization constant of the Mallows posterior (given partial evidence v). The algorithm is specified in detail in Alg. 4.

Exploiting Proposition 11, we can show:

Theorem 16. *The Markov chain induced by* MMP *is ergodic on the class of states (rankings)* $\mathsf{C}(v)$.

Proof. Note that the acceptance ratio as given in Eq. 5.23 is always positive. The proposal distribution AMP draws rankings that are independent of previous rankings and by Proposition 11, its support is $\mathsf{C}(v)$. Hence, for any $r' \in \mathsf{C}(v)$, MMP has positive probability of making a transition to any ranking in $\mathsf{C}(v)$—thus establishing that $\mathsf{C}(v)$ is a recurrent class—including itself—implying aperiodicity. □

Thus, along with the detailed balance property of Metropolis, we have that the steady state distribution of MMP is exactly the Mallows posterior $P_v(r)$.

5.4.3 Sampling Mallows Mixture Posterior

Extending the GRIM, AMP and MMP algorithms to sampling from a mixture of Mallows models is straightforward. Recall the mixture posterior:

$$P(r, \mathbf{z}|v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}) = \frac{P(v|r, \alpha)P(r|\mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}|\boldsymbol{\pi})}{\sum_{\mathbf{z}} \sum_{r \in \mathsf{C}} P(v|r, \alpha)P(r|\mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}|\boldsymbol{\pi})}.$$

We use Gibbs sampling to alternate between r and $\mathbf{z}$, since the posterior does not factor in a way that permits us to draw samples exactly by sampling one variable, then conditionally sampling another. We initialize the process with some $\mathbf{z}^{(0)}$ and $r^{(0)}$, then repeatedly sample $\mathbf{z}$ conditional on r, and r conditional on $\mathbf{z}$. For the tth sample, $\mathbf{z}^{(t)}$ is drawn from a multinomial with K outcomes:

$$\begin{aligned} P(\mathbf{z} : z_k = 1|r^{(t-1)}, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}) &= \frac{P(r^{(t-1)}|\mathbf{z}, \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}|\boldsymbol{\pi})}{\sum_{\mathbf{z}'} P(r^{(t-1)}|\mathbf{z}', \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}'|\boldsymbol{\pi})} \\ &= \frac{\phi_k^{d(r^{(t-1)}, \sigma^{(k)})}\pi_k}{\sum_{k'=1}^{K} \phi_{k'}^{d(r^{(t-1)}, \sigma^{(k')})}\pi_{k'}}. \end{aligned}$$

To sample $r^{(t)}$ given $\mathbf{z}^t$, we use:

$$P(r|\mathbf{z}^{(t)}, v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}) = \frac{P(v|r)P(r|\mathbf{z}^{(t)}, \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}^{(t)}|\boldsymbol{\pi})}{\sum_{r' \in \mathsf{C}} P(v|r')P(r'|\mathbf{z}^{(t)}, \boldsymbol{\sigma}, \boldsymbol{\phi})P(\mathbf{z}^{(t)}|\boldsymbol{\pi})}. \tag{5.24}$$

Note that the term $P(\mathbf{z}^{(t)}|\boldsymbol{\pi})$ in the numerator and denominator cancels, and the missing completely at random assumption (see Eq. 5.9) implies $P(v|r) = \mathbf{1}[r \in \mathsf{C}(v)]f(v)$, where f is a function independent of r. Thus Eq. 5.24 becomes Eq. 5.12 (conditioned on parameters $\sigma^{(k)}$, ϕ_k). This is exactly the Mallows posterior sampling problem addressed in the previous section. Combining Gibbs sampling with sampling from a single component gives the overall SP algorithm, which is detailed in Algorithm 5. We note that this sampler is described using either MMP to exactly sample rankings (given the sampled mixture component) or AMP to allow more tractable, but approximate, sampling of rankings (see Line 5). In our experiments, we find that AMP works well within this Gibbs sampler.

5.5 EM Learning Algorithm for Mallows Mixtures

Armed with the sampling algorithms derived from GRIM, we now turn to maximum likelihood learning of the parameters $\boldsymbol{\pi}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\phi}$ of a Mallows mixture using the *ex-*

Algorithm 5 SP: Sample Mallows Mixture Posterior using Gibbs

Input: $v, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}$, number of steps T

1: Initialize $r^{(0)}$ (e.g., topological sort on v)
2: **for** $t = 1..T$ **do**
3: $\quad \mathbf{z}^{(t)} \sim P(\cdot|r^{(t-1)}, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}) \propto \phi_k^{d(r^{(t-1)}, \sigma^{(k)})} \pi_k$
4: $\quad$ Suppose $\mathbf{z}^{(t)}$ is indicator for kth component.
5: $\quad r^{(t)} \leftarrow$ AMP or MMP$(v, \sigma^{(k)}, \phi_k)$
6: **end for**

Output: $(\mathbf{z}^{(T)}, r^{(T)})$

pectation maximization (EM) algorithm. Before detailing our EM algorithm, we first consider the evaluation of the Mallows mixture log-likelihood in Section 5.5.1, which can be used to select the number of mixture components, or to test EM learning convergence. We then review the EM algorithm in Section 5.5.2 before detailing the steps of our EM learning procedure for Mallows mixture models in Section 5.5.3. In Section 5.5.4 we analyze the running time of our learning algorithm and suggest several ways to improve its performance.

5.5.1 Evaluating Log-Likelihood

The log-likelihood in our mixture model is:

$$\mathcal{L}_\alpha(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} \mid \mathbf{v}) = \sum_{\ell \in N} \ln \left[\sum_{\mathbf{z}_\ell} \sum_{r_\ell \in \mathsf{C}} P(v_\ell|r_\ell) P(r_\ell|\mathbf{z}_\ell, \boldsymbol{\sigma}, \boldsymbol{\phi}) P(\mathbf{z}_\ell|\boldsymbol{\pi}) \right]. \tag{5.25}$$

This can be rewritten as:

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} \mid \mathbf{v}) &= \sum_{\ell \in N} \ln \left[\sum_{k=1}^{K} \sum_{r_\ell \in \mathsf{C}(v_\ell)} \pi_k P(r_\ell|\sigma^{(k)}, \phi_k) \alpha^{|v_\ell|} (1-\alpha)^{\binom{m}{2} - |v_\ell|} \right] \\ &= \sum_{\ell \in N} \ln \left[\sum_{k=1}^{K} \sum_{r_\ell \in \mathsf{C}(v_\ell)} \pi_k P(r_\ell|\sigma^{(k)}, \phi_k) \right] + \ln \left[\alpha^{|v_\ell|} (1-\alpha)^{\binom{m}{2} - |v_\ell|} \right]. \end{aligned}$$

Note that the latter term involving α is decoupled from the other parameters, and in fact its maximum likelihood estimate is $\alpha^* = \sum_{\ell \in N} 2|v_\ell|/(nm(m-1))$. Since we are only interested in the log-likelihood as a function of the other parameters, we can ignore this

additive constant and focus on

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi} \mid \mathbf{v}) = \sum_{\ell \in N} \ln \left[\sum_{k=1}^{K} \sum_{r_\ell \in \mathsf{C}(v_\ell)} \frac{\pi_k \phi_k^{d(r_\ell, \sigma^{(k)})}}{Z_k} \right], \tag{5.26}$$

where Z_k is the Mallows normalization constant. Unfortunately, evaluating this term is provably hard.

Theorem 17. *Let $\mathbf{v} = (v_1, \ldots, v_n)$ be a profile of partial preferences. Computing the log-likelihood $\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}|\mathbf{v})$ is #P-hard.*

Proof. We reduce the problem of counting the number of linear extensions of a partial order to this problem (see Theorem 13, due to Brightwell and Winkler [20]). Let v be a partial order for which we wish to count its linear extensions. We encode the input to log-likelihood computation as follows: let $\mathbf{v} = (v)$, $K = 1$ with $\phi = 1$, and let σ be an arbitrary ranking. We have $\mathcal{L} = \mathcal{L}(\pi, \sigma, \phi|\mathbf{v}) = \ln \sum_{r \in \mathsf{C}(v)} 1/m!$. Thus we can recover the number of linear extensions by computing $\exp(\mathcal{L}) \cdot m!$. That this can be accomplished in polynomial time can be seen by noting that $\mathcal{L}$ is polynomial in m and we can use the power series expansion $\sum_{i \geq 0} \mathcal{L}^i m!/i!$, where we can truncate the series after a polynomial number of steps, after which the terms in the expansion no longer impact the integer portion of the solution (number of extensions). □

Given the computational difficulty of evaluating the log-likelihood exactly, we consider approximations. We can rewrite the log-likelihood as:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}|\mathbf{v}) = \sum_{\ell \in N} \ln \left[\sum_{k=1}^{K} \pi_k \mathop{\mathbb{E}}_{P(r|\sigma^{(k)}, \phi_k)} \mathbf{1}[r \in \mathsf{C}(v)] \right],$$

and estimate the inner expectations by sampling from the Mallows model $P(r|\sigma^{(k)}, \phi_k)$. However, this can require exponential sample complexity in the worst-case (e.g., if $K = 1$ and v is far from σ, i.e., $d(v, \sigma)$ is large, then to ensure v is in the sample requires a sample set of exponential size in expectation). But we can rewrite the summation inside the log as:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}|\mathbf{v}) \;=\; \sum_{\ell \in N} \ln \left[\sum_{k=1}^{K} \frac{\pi_k}{Z_k} \sum_{r \in \mathsf{C}(v_\ell)} \phi_k^{d(r, \sigma^{(k)})} \right],$$

and evaluate $\sum_{r\in\mathsf{C}(v_\ell)} \phi_k^{d(r,\sigma^{(k)})}$ using importance sampling:

$$\sum_{r\in\mathsf{C}(v_\ell)} \phi_k^{d(r,\sigma^{(k)})} = \mathop{\mathbb{E}}_{r\sim\hat{P}_{v_\ell}} \left[\frac{\phi_k^{d(r,\sigma^{(k)})}}{\hat{P}_{v_\ell}(r|\sigma^{(k)},\phi_k)} \right]. \tag{5.27}$$

We generate samples $r_{\ell k}^{(1)},\ldots,r_{\ell k}^{(T)}$ with $\mathsf{AMP}(v_\ell,\sigma^{(k)},\phi_k)$ for $\ell \le n$ and $k \le K$, then substitute $\hat{P}_v$ from Eq. 5.15 into Eq. 5.27 to obtain:

$$\sum_{\ell\in N} \ln \left[\sum_{k=1}^{K} \frac{\pi_k}{Z_k} \frac{1}{T} \sum_{t=1}^{T} \prod_{i=1}^{m} \sum_{j=i-h_i^{(\ell kt)}}^{i-l_i^{(\ell kt)}} \phi_k^j \right],$$

where $h_i^{(\ell kt)}$ and $l_i^{(\ell kt)}$ are defined in Eq. 5.14 and 5.13, and defined with respect to $r_{\ell k}^{(t)}$, $\sigma^{(k)}$, and v_ℓ. We can simplify the expression inside the log and derive the estimate:

$$\hat{\mathcal{L}}(\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi} \mid \mathbf{v}) = \sum_{\ell\in N} \ln \left[\frac{1}{T} \sum_{k=1}^{K} \sum_{t=1}^{T} \pi_k \cdot \begin{cases} \frac{1}{m!} \prod_{i=1}^{m} (h_i^{(\ell kt)} - l_i^{(\ell kt)} + 1) & \text{if } \phi_k = 1, \\ \phi_k^{\sum_{i=1}^{m} i - h_i^{(\ell kt)}} \prod_{i=1}^{m} \frac{1-\phi_k^{h_i^{(\ell kt)} - l_i^{(\ell kt)} + 1}}{1-\phi_k^i} & \text{otherwise.} \end{cases} \right] \tag{5.28}$$

As a matter of practical implementation, to ensure the sum of terms inside the log do not evaluate to zero (as it may be too small to be represented using common floating point standards), we observe that given numbers a and b with $a > b > 0$, $\ln(a+b) = \ln(a) + \ln(1+b/a)$. Thus even if a and b are too small to be represented as floating point data types, we still obtain good approximations if $\ln(a)$ can be readily evaluated. This same technique can be used to ensure numerical stability.

5.5.2 The EM Algorithm

A popular approach to maximum likelihood estimation is the *expectation maximization (EM)* algorithm [40]. It is applied to probabilistic models in which a set of parameters θ determine the values of random variables, but observed data is available for only some of these variables. Let v denote the observed variables, and h the remaining unobserved (hidden or latent). In our model, we have $\theta = (\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi},\alpha)$, while v consists of a set of pairwise comparisons and $h = (\mathbf{z}, r)$ consists of the mixture-component assignment and its underlying complete preference ranking. EM is effectively a local search algorithm,

which alternates between two steps. The *E-step* computes a posterior distribution over the hidden variables given the observed variables and a current estimate $\tilde{\theta}$ of the model parameters:

$$\textbf{E-Step:} \quad P(h|v,\tilde{\theta}).$$

The *M-step* computes, as its new estimate, those model parameters θ that maximize the expected value (w.r.t. $\tilde{\theta}$) of the log-likelihood (using the posterior computed in the E-step):

$$\textbf{M-step:} \quad \max_{\theta} \mathop{\mathbb{E}}_{P(h|v,\tilde{\theta})} \ln P(h,v|\theta).$$

These steps are iterated until convergence. Indeed, EM converges and gives a locally optimal solution, since each iteration of EM will increase the log-likelihood. In general one does not need to maximize the log-likelihood in the M-step, but simply increase it. An important variation of EM called *Monte Carlo EM* is used when the posterior in the E-step is hard to compute (e.g., when dealing with large discrete event spaces, such as rankings). In Monte Carlo EM, ones samples from the posterior in the E-step, and in the M-step simply optimizes the choice of parameters with respect to the empirical (sample) expectation.

5.5.3 Monte Carlo EM for Mallows Mixtures

Learning a Mallows mixture is challenging, since even evaluating its log-likelihood is #P-hard. A straightforward application of EM yields the following algorithm:

Initialization. Initialize values for $\boldsymbol{\pi}^{\text{old}}$, $\boldsymbol{\sigma}^{\text{old}}$, and $\boldsymbol{\phi}^{\text{old}}$.
E-step. Compute/estimate the posterior $P(\mathbf{z}_\ell, r_\ell|v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}})$ for all $\ell \in N$.
M-step. Compute model parameters that maximize expected log-likelihood:

$$\begin{aligned}\boldsymbol{\pi}^{\text{new}}, \boldsymbol{\sigma}^{\text{new}}, \boldsymbol{\phi}^{\text{new}} &= \operatorname*{argmax}_{\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi}} \sum_{\ell \in N} \mathop{\mathbb{E}}_{P(r_\ell,\mathbf{z}_\ell|v_\ell,\boldsymbol{\pi}^{\text{old}},\boldsymbol{\sigma}^{\text{old}},\boldsymbol{\phi}^{\text{old}})} \left[\ln P(v_\ell, r_\ell, \mathbf{z}_\ell|\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi})\right] \\ &= \operatorname*{argmax}_{\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi}} \sum_{\ell \in N} \sum_{\mathbf{z}_\ell} \sum_{r_\ell \in \mathsf{C}} P(r_\ell, \mathbf{z}_\ell|v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}) \ln P(v_\ell, r_\ell, \mathbf{z}_\ell|\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi}).\end{aligned}$$

Exact estimation in the E-step and optimization in the M-step is of course difficult due to the intractability of the Mallows posterior. Hence we resort to *Monte Carlo EM* and exploit our sampling methods to render EM tractable as follows. We initialize the parameters with values $\boldsymbol{\pi}^{\text{old}}$, $\boldsymbol{\sigma}^{\text{old}}$, and $\boldsymbol{\phi}^{\text{old}}$. For the E-step, instead of working directly with the posterior, we use GRIM-based Gibbs sampling (see Section 5.4.3) to obtain samples $(\mathbf{z}_\ell^{(t)}, r_\ell^{(t)})_{t=1}^T$ from the posteriors $P(r_\ell, \mathbf{z}_\ell|v_\ell, \boldsymbol{\pi}^{\text{old}}, \boldsymbol{\sigma}^{\text{old}}, \boldsymbol{\phi}^{\text{old}})$ of each agent $\ell \leq n$.

We note once again that Gibbs sampling may use either approximate AMP or the full-fledged MCMC MMP to generate rankings.

In the M-step, we maximize the expected log-likelihood using the *empirical expectation* with respect to the generated samples:

$$\boldsymbol{\pi}^{\text{new}}, \boldsymbol{\sigma}^{\text{new}}, \boldsymbol{\phi}^{\text{new}} = \operatorname*{argmax}_{\boldsymbol{\pi},\boldsymbol{\sigma},\boldsymbol{\phi}} \sum_{\ell=1}^{n} \frac{1}{T} \sum_{t=1}^{T} \ln P(v_\ell, r_\ell^{(t)}, \mathbf{z}_\ell^{(t)} | \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi}). \tag{5.29}$$

We show below in Theorem 18 that we can perform this maximization by adjusting the three (sets of) parameters in sequence—specifically, if the parameters are maximized in the order $\boldsymbol{\pi}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\phi}$ (and the first two can be maximized independently), this provides a globally optimal solution for the M-step (i.e., the solution obtained by optimizing parameters simultaneously). However, optimization of $\boldsymbol{\sigma}$, in particular, is NP-hard (as we discuss below), so we use a local search heuristic to approximate the choice of reference rankings in the M-step. We now detail the steps involved in the M-step optimization.

Somewhat abusing notation, let indicator vector $\mathbf{z}_\ell^{(t)}$ denote the mixture component to which the tth sample derived from preference ℓ belongs. We partition the collection of *all agent samples* (over all ℓ) into such classes: let $S_k = (\rho_{k1}, \ldots, \rho_{kj_k})$ be the sub-sample of the rankings $r_\ell^{(t)}$, over all $\ell \in N, t \in [T]$, that are drawn from the kth component of the mixture model, i.e., where $\mathbf{z}_\ell^{(t)} = k$. Note that $j_1 + \cdots + j_K = nT$. We can rewrite the objective in the M-step as:

$$\frac{1}{T} \sum_{k=1}^{K} \sum_{i=1}^{j_k} \ln P(v_{\ell(k,i)} | \rho_{ki}) P(\rho_{ki} | \sigma^{(k)}, \phi_k) P(k | \pi_k),$$

where $\ell(k,i)$ is the agent in sample $\rho_{k,i}$. We ignore $\ln P(v_{\ell(k,i)} | \rho_{ki})$, which only impacts α; and we know $\rho_{ki} \in \mathsf{C}(v_{\ell(k,i)})$. Thus, we can rewrite the objective as:

$$\sum_{k=1}^{K} \sum_{i=1}^{j_k} \left[\ln \pi_k + d(\rho_{ki}, \sigma^{(k)}) \ln \phi_k - \sum_{w=1}^{m} \ln \frac{1 - \phi_k^w}{1 - \phi_k} \right]. \tag{5.30}$$

where the last summation is the log of the Mallows normalization term.

Optimizing $\boldsymbol{\pi}$

We apply the method of Lagrange multipliers. The Lagrangian $L = (\sum_{k=1}^{K} \sum_{i=1}^{j_k} \ln \pi_k) + \lambda(\pi_1 + \cdots + \pi_K - 1)$, where we have removed irrelevant terms of the objective not involving $\boldsymbol{\pi}$. Taking the gradient, setting to zero and solving the system of equations $\nabla_{\boldsymbol{\pi},\lambda} L = 0$,

we obtain

$$\pi_k = \frac{j_k}{nT},\ \forall k \leq K. \tag{5.31}$$

Optimizing $\boldsymbol{\sigma}$

The only term involving $\boldsymbol{\sigma}$ in Eq. 5.30 is $\sum_{k=1}^{K}\sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)}) \ln \phi_k$. Since $\ln \phi_k$ is a negative scaling factor, and we can optimize the reference rankings $\sigma^{(k)}$ for each mixture component independently, we obtain:

$$\sigma^{(k)*} = \operatorname*{argmin}_{\sigma^{(k)}} \sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)}). \tag{5.32}$$

Optimizing the choice of reference ranking $\sigma^{(k)}$ within a mixture component requires computation of the Kemeny ranking with respect to the rankings in S_k. This is, unfortunately, an NP-hard problem [8]. To maintain tractability, we exploit the notion of *local Kemenization* [45]: instead of optimizing the ranking, we compute a locally optimal $\sigma^{(k)}$, in which swapping any two *adjacent* alternatives in $\sigma^{(k)}$ does not reduce the sum of distances in the Kemeny objective. While this may not result in optimal rankings, it has been shown to be extremely effective experimentally [45, 24].

We detail our local Kemenization algorithm in Algorithm 6, a slight adaptation of Algorithm 1 from Chapter 2. It works by first initializing the new ranking $\sigma^{(k)}$ to that from the previous EM iteration, $\sigma^{\text{old},(k)}$. Then, for each alternative x, starting with those at the top of the ranking and moving downwards, we evaluate swaps of x with the element above it, say y, and proceeding with the swap if the majority of rankings in S_k prefer x over y. This proceeds until the first potential swap of x fails (at which point we move on to the next alternative). This results in a locally optimal ranking [45]. Note we need not store all rankings in S_k; we require only its *pairwise tournament graph*, which is a complete directed graph with vertices corresponding to the alternatives A and the weight of each edge $x \to y$ set to be $c_{xy} = |\{\rho \in S_k : y \succ_\rho x\}|$. Here c_{xy} is the "cost" of placing x above y.

Optimizing $\boldsymbol{\phi}$

When optimizing $\boldsymbol{\phi}$ in Eq. 5.30, the objective decomposes into a sum that permits independent optimization of each ϕ_k. Exact optimization of ϕ_k is difficult; however, we can

Algorithm 6 LocalKemeny for σ_k

Input: $S_k = (\rho_{k1}, \ldots, \rho_{kj_k})$

$\sigma \leftarrow \sigma_k^{\text{old}}$
Compute pairwise tournament graph:
for all pair $(x, y) : x, y \in A$ and $x \neq y$ **do**
  $c_{xy} = |\{\rho \in S_k : y \succ_\rho x\}|$.
end for
$d \leftarrow \sum_{\{x,y\} \,:\, x \succ_{\sigma^{(k)}} y} c_{xy}$
for $i = 2..m$ **do**
  $x \leftarrow$ alternative in ith rank of σ
  for $j = i - 1..1$ **do**
    $y \leftarrow$ alternative in jth rank of σ
    if $c_{xy} < c_{yx}$ **then**
      Swap x with y
      $d \leftarrow d - c_{xy} + c_{yx}$
    else
      quit this loop
    end if
  end for
end for

Output: σ, Kemeny cost d

use gradient ascent with:

$$\frac{\partial \text{ (Eq. 5.30)}}{\partial \phi_k} = \frac{d(S_k, \sigma^{(k)})}{\phi_k} - j_k \sum_{i=1}^{m} \frac{[(i-1)\phi_k - i]\phi_k^{i-1} + 1}{(1 - \phi_k^i)(1 - \phi_k)},$$

where $d(S_k, \sigma^{(k)}) = \sum_{i=1}^{j_k} d(\rho_{ki}, \sigma^{(k)})$ is the Kemeny objective, which we obtain after running LocalKemeny.

Theorem 18. *Let $\boldsymbol{\pi}^*$ be given by Eq. 5.31, $\boldsymbol{\sigma}^*$ be given by Eq. 5.32, and $\boldsymbol{\phi}^*$ be the optimal $\boldsymbol{\phi}$ in Eq. 5.30 where $\boldsymbol{\pi}$ is replaced with $\boldsymbol{\pi}^*$ and $\boldsymbol{\sigma}$ is replaced with $\boldsymbol{\sigma}^*$. Then $\boldsymbol{\pi}^*$, $\boldsymbol{\sigma}^*$ and $\boldsymbol{\phi}^*$ is a globally optimal solution to Eq. 5.29.*

Proof. Regardless of the values of $\boldsymbol{\sigma}$ and $\boldsymbol{\phi}$, $\boldsymbol{\pi}$ is optimized by Eq. 5.31 (see our analysis above), giving the optimal solution. It is also easy to see that the optimal reference rankings $\boldsymbol{\sigma}$ are the Kemeny rankings corresponding to ranking sets $S_1, \ldots, S_K$, respectively, independent of the value of $\boldsymbol{\phi}$. Finally, if we substitute the optimal values $\boldsymbol{\pi}^*$ and $\boldsymbol{\sigma}^*$ into Eq. 5.30, its optimal solution $\boldsymbol{\phi}^*$ forms part of the optimal solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*, \boldsymbol{\phi}^*)$ to Eq. 5.29. □

It isn't difficult to see that a "locally optimal" pair $(\boldsymbol{\sigma}, \boldsymbol{\phi})$ obtained by optimizing $\boldsymbol{\sigma}$

first, then $\boldsymbol{\phi}$ is a locally optimal pair for Eq. 5.29. Hence the resulting EM estimates are also locally optimal with respect to the likelihood [96]. While no approximation bounds can be given, this lends some support to the optimization approach we adopt. To test the convergence of EM, one can test the convergence of the parameters (use Kendall-tau distance to measure $\boldsymbol{\sigma}$ against that of the previous iteration). One can also measure whether the log-likelihood is converging.

To reduce problems with local maxima, we initialize the mixture parameters using a K-means clustering approach where distances are measured using Kendall-tau rather than the usual squared Euclidean distance. One can use a modified version of Lloyd's 1982 method for K-means, where the "centroid" (pertaining to Lloyd's method) of a set of rankings can is simply its Kemeny ranking.

5.5.4 Complexity of EM Steps

We analyze the running time of one iteration of our EM approach. In the E-step, we sample variables $(\mathbf{z}, r)$. We need not store the ranking r for the component corresponding to $\mathbf{z}$, since in the M-step we do not need the actual rankings in S_k, but only its pairwise tournament graph. Hence we need only update the tournament graph corresponding to component $\mathbf{z}$ with sample r, which takes $O(m^2)$ time. When sampling r, let T_{Metro} be the number of Metropolis steps before using the next sample. Each draw of r from AMP requires $O(m^2)$ time. Sampling $\mathbf{z}$ requires $O(Km \log m)$ time since Kendall-tau distance can be computed in $O(m \log m)$ time. Let T_{Gibbs} be the number of Gibbs sampling steps run Gibbs before outputting a sample and suppose we restart Gibbs after each such sample. Suppose also we draw T_P posterior samples for each data point v_ℓ. Then the E-step takes $O(nT_P T_{\text{Gibbs}}(T_{\text{Metro}} m^2 + Km \log m))$ time. In practice, one can chose a very small number of samples, and run relatively few steps, when running the MCMC methods. Indeed, in our experiments below, we don't use MMP within the Gibbs sampler, but instead use AMP directly (this can be viewed as running Metropolis for a single step); we discuss this further below. In principle, posterior sampling can be executed in parallel, with multiple processors handling the sampling and tournament graph updates for disjoint subsets of the data v_ℓ, with the results from different processors merged into the K tournament graphs.

For the M-step, updating $\boldsymbol{\pi}$ takes constant time, while updating the component reference rankings $\boldsymbol{\sigma}$ takes $O(Km^2)$ time. Optimizing $\boldsymbol{\phi}$ can also be realized effectively, for instance, by using gradient ascent and bounding number of iterations. Hence the M-step requires $O(Km^2)$ time. Space complexity is dominated by the size of the K tournament

graphs, hence is $O(Km^2)$.

Various techniques can be used to speed up computation from a practical perspective. Instead of storing the tournament graphs, which require quadratic memory, one can instead approximate the Kemeny ranking for any component using the *Borda count* to rank alternatives, which is a 5-approximation to Kemeny [38], and often provides much better approximations in practice. If using Borda, when generating a complete ranking r in the posterior-sampling step (E-step) belonging to component k, one need only to update the Borda scores of all alternatives within component k; in the M-step we simply rank alternatives (within each component) according to their sampled Borda scores. We still need the Kemeny distance between the resulting Borda ranking and the sampled rankings, but this can be approximated by re-running the E-step and evaluating the Kendall-tau distance in an online fashion. One might also consider using Spearman footrule distance, which can be computed in $O(m)$ time rather than $O(m \log m)$ as in Kendall-tau, since it is 2-approximation to the Kemeny distance [41].

5.6 Empirical Evaluation

We perform a series of experiments to validate the efficacy of our sampling and learning algorithms, to discover interesting properties of the learned mixture models on several popular datasets, and to evaluate the predictive power of our learned models to help predict missing preferences. We first assess the quality of our GRIM-based posterior sampling method AMP, measuring its accuracy relative to the true Mallows posterior. We then measure the approximation quality of our Monte Carlo algorithm for evaluating the Mallows mixture log-likelihood. Next we apply our EM algorithm to learn mixture models using several data sets: synthetically generated datasets, a Movielens ratings dataset (with large m), and a sushi preference dataset. The synthetic data experiments confirm the effectiveness of our EM algorithm while also revealing insights on how the size of preference data (either n or α) impacts learning. We also remark on some of its connections to crowdsourcing. Finally we assess the predictive accuracy of the learned models by conditioning on partial preference information and inferring the probability of the missing pairwise comparison preferences. In all experiments, we use Eq. 5.26 to measure log-likelihood.

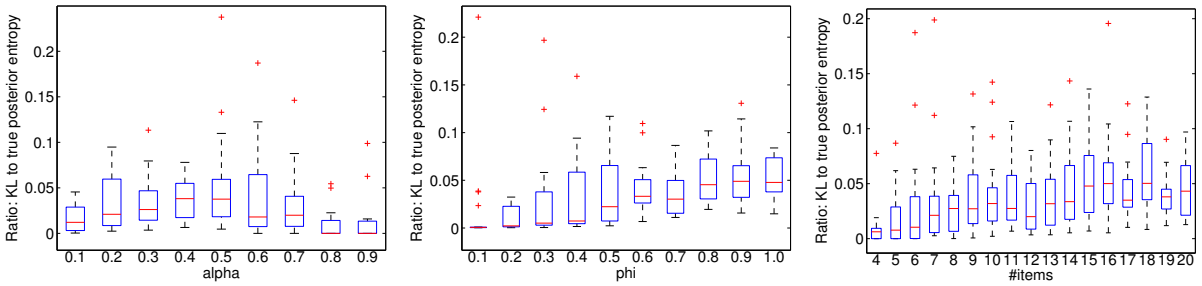

Figure 5.3: Comparing the posterior generated by AMP to the true Mallows posterior: normalized KL-divergence. The box-and-whisker plots have boxes shown the 25-75 percentile range over 20 runs, with the line inside box indicating the median, and the '+' symbols outliers. From left to right: **Plot 1**: Varying α, while fixing $\phi = 0.5$, $m = 10$. **Plot 2**: Varying ϕ, while fixing $\alpha = 0.2$, $m = 10$. **Plot 3**: Varying m, fixing $\phi = 0.5$ and for $m \leq 13$, $\alpha = 0.2$, for $m > 13$, $\alpha = 0.5$.

5.6.1 Sampling Quality

We first assess how well AMP approximates the true Mallows posterior P_v using randomly generated (synthetic) data. We vary parameters m, ϕ and α, while fixing a canonical reference ranking $\sigma = (1, 2, \ldots, m)$. For each parameter setting, we generate 20 preferences v (e.g., the partial preferences of 20 agents) using our mixture model (see Sec 5.2.3 and Eq. 5.9 and 5.10), and evaluate the exact KL-divergence of P_v with respect to $\hat{P}_v$.[2] This divergence is normalized by the entropy of P_v, since, when increasing m, KL-divergence and entropy both increase. Results are shown in Figure 5.3, with fixed and varying parameters for all three plots described in the caption. These results indicate that AMP approximates the posterior very well, with average normalized KL error ranging from 1–5%, across the parameter ranges tested. For this reason, and given its much better running time, we use AMP within our evaluation of EM rather than using the asymptotically exact sampler MMP.

5.6.2 Evaluating Log-Likelihood

In Section 5.5.1 we showed the #P-hardness of evaluating the log-likelihood and derived a Monte Carlo estimator that uses the AMP sampler. We evaluate the quality of the approximation produced by this estimator in this section. We vary three parameters to generate three experiments: (a) the number of alternatives m; (b) the number of mixture components K; and (c) the number of samples T per agent and per component (Eq. 5.28).

[2] To compute KL-divergence, we need only consider consistent completions of our partial preferences. This set of rankings usually has size much smaller than $m!$, and can be enumerated by modifying the topological sort algorithm.

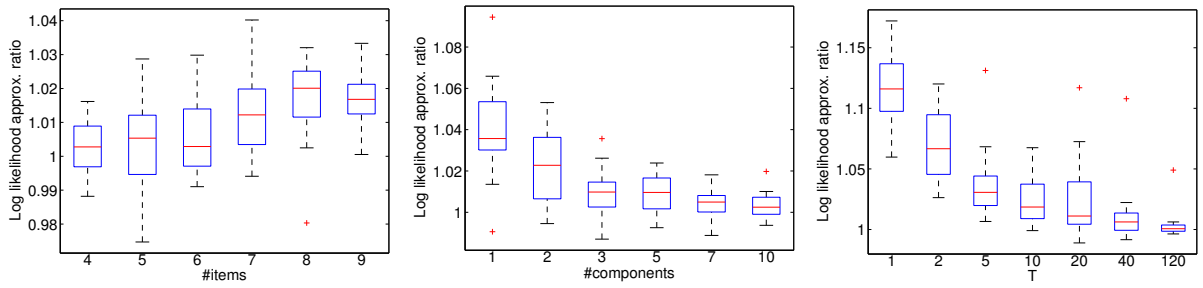

Figure 5.4: Comparing the ratio of the true log-likelihood to its Monte Carlo approximation. 20 instances are run per parameter setting. From left to right: **Plot 1**: Varying m, while fixing $T = 5$. **Plot 2**: Varying K while fixing $T = 5$. **Plot 3**: Varying T. Other parameter values are described in the text.

In all experiments, we fix the number of agents (i.e., the number of input preferences) at $n = 50$.

In setting (a) (varying m), we generate v from a mixture model with $K = 3$ and $\boldsymbol{\pi} = (1/3, 1/3, 1/3)$, $\boldsymbol{\phi} = (1/2, 1/2, 1/2)$ and $\alpha = 0.2$. Each $\boldsymbol{\sigma}^k$ $(k \leq K)$ is drawn uniformly at random from C.

In setting (b) (varying K), we generate v from a mixture model with K components, where $m = 8$, $\boldsymbol{\pi} = (1/K, \ldots, 1/K)$, $\boldsymbol{\phi} = (1/2, \ldots, 1/2)$ and $\alpha = 0.2$. Again $\boldsymbol{\sigma}$ drawn uniformly at random as in setting (a).

In setting (c) (varying T), parameters are $K = 1$, $m = 8$, σ chosen uniformly at random, $\phi = 0.5$ and $\alpha = 0.2$.

The parameters for which we evaluated the log-likelihood are generated as follows: mixture weights $\boldsymbol{\pi}$ are sampled from a "uniform" Dirichlet distribution with a parameter vector (i.e., equivalent sample size counts) consisting of K 5s. The reference rankings $\boldsymbol{\sigma}$ were drawn uniformly at random, and ϕ is drawn uniformly at random from interval $(0, 1)$.

The results for all three settings are shown in Figure 5.4. Overall we see that the Monte Carlo approximation is very good, and improves significantly while reducing variance as we increase the sample size for each agent's log-likelihood (as captured by $K \cdot T$). Increasing m slightly degrades approximation quality, although it offers excellent estimates across the entire range of tested values.

5.6.3 EM Mixture Learning

We now evaluate our EM mixture-learning algorithms on the synthetic, Sushi and Movielens datasets.

5.6.3.1 Synthetic Data

Having empirically established that AMP provides good approximations to the true posterior, and that the log-likelihood can be closely approximated by importance sampling, we now evaluate how effective our EM algorithm is at recovering parameters in a controlled setting, using synthetic data generated from models with known parameters. We emphasize that the following experiments all use AMP within the Gibbs sampler in the E-step of Monte Carlo EM, rather than the MCMC algorithm MMP, given the approximation quality of AMP as well as its much better tractability.

We perform four experiments in which we vary: (a) α, the (expected) fraction of pairwise comparisons revealed from each preference; (b) the number of alternatives m; (c) the number of mixture components K; and (d) the number of agent preferences (data set size). In each experiment, we generate random model parameters as follows: $\boldsymbol{\pi}$ is drawn from a Dirichlet distribution with a uniform parameter vector of 5s; $\boldsymbol{\sigma}$ is drawn uniformly at random; and $\boldsymbol{\phi}$ values are drawn uniformly at random from $[0.2, 0.8]$. Training data is generated using our probabilistic model with these parameters. When varying the single parameter for each experiment, we fix the other three, with fixed values: $\alpha = 0.2$, $m = 20$, $K = 3$ and $n = 50 \times K$. We analyze the performance of EM by (approximately) evaluating the ratio of the log-likelihood of the learned parameters to that of the true model parameters $(\boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\phi})$ on test data (preferences) generated from the true model—we set $n_{test} = n$ and $\alpha_{test} = 1$.

Results are shown in Figure 5.5 and provide some interesting insights. First they suggest that learning is more effective when either of α or n is larger (i.e., when we have more preference data for training). We also see that learning performance degrades when we increase the number of mixture components—this is hardly surprising, since there is less data per component as we increase K. Finally, learning improves as m increases for fixed values of α. This holds because the transitive closure for larger m tends to offer more preference information. For instance, $a_1 \succ a_2 \succ a_3 \succ a_4 \succ a_5 \succ a_6$ provides 5 comparisons, and corresponds to 1/9 of all comparisons when $m = 10$, while leaving many comparisons unavailable, even after taking its transitive closure. By contrast, $a_1 \succ a_2 \succ \cdots \succ a_{100}$ has 99 comparisons which is only 1/50 of all comparisons when $m = 100$; but its transitive closure is a complete ranking.

These observations have interesting implications when considering information elicitation via "wisdom of the crowds." When estimating a single objective ranking (i.e., $K = 1$), the amount of data needed for reliable estimation can be obtained by either increasing α (the average number of pairwise comparisons revealed per agent) and decreasing n (the number of agents queried) or by increasing n and decreasing α. In other

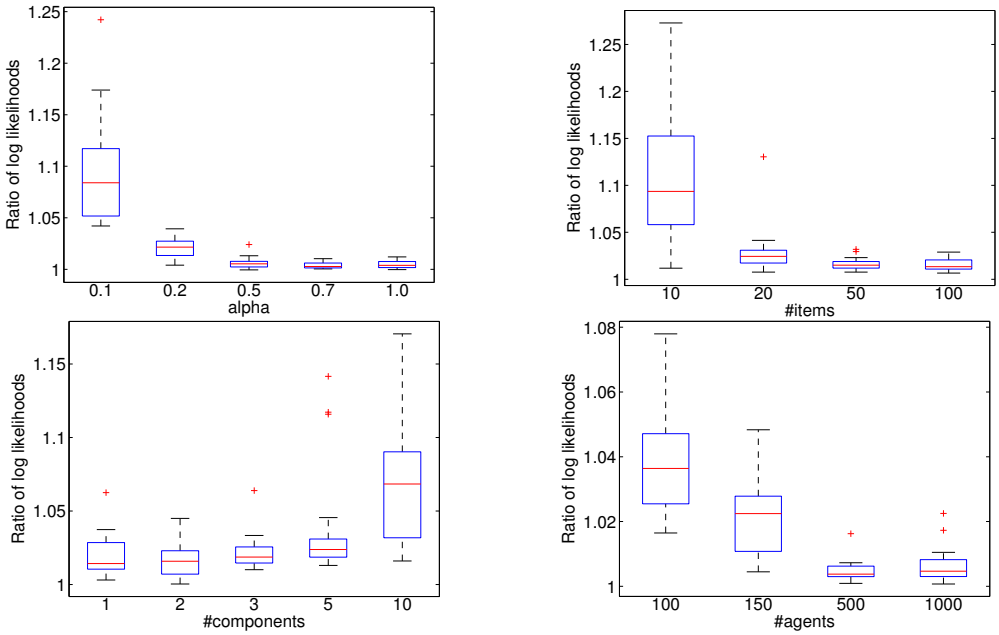

Figure 5.5: Performance of EM on synthetic data. Each plot shows the ratio of the log-likelihood of learned parameters to those of the true model parameters. Each parameter setting averages results of 20 instances. Log-likelihoods are approximated as in Section 5.5.1 with $T = 10$. Other parameter settings are described in the text.

words, one can obtain the same "effective" data by either asking more agents about their objective assessments while decreasing the number of questions per agent, as asking fewer agents to respond, but demanding more pairwise assessments per agent.

5.6.3.2 Sushi Data

The Sushi dataset consists of 5000 complete rankings over 10 varieties of sushi indicating sushi preferences [70]. We used 3500 preferences for training and 1500 for validation. We ran EM experiments by generating revealed pairwise comparisons for training with various probabilities α. To mitigate issues with local maxima, we ran EM ten times (more than is necessary) for each instance. Figure 5.6 shows that, even without complete preferences, EM learns well even with only 30-50% of all paired comparisons, though it degrades significantly at 20%, in part because only 10 alternatives are ranked (still performance at 20% is good when $K = 1, 2$). With $K = 6$ components, a good fit is found when training on complete preferences: Table 5.2 shows the learned clusters (all with reasonably low dispersion), illustrating interesting patterns (e.g., fatty tuna is strongly preferred by all but one group; a strong correlation exists across groups in preference/dispreference for salmon roe and sea urchin, which are "atypical fish"; and cucumber roll is consistently dispreferred).

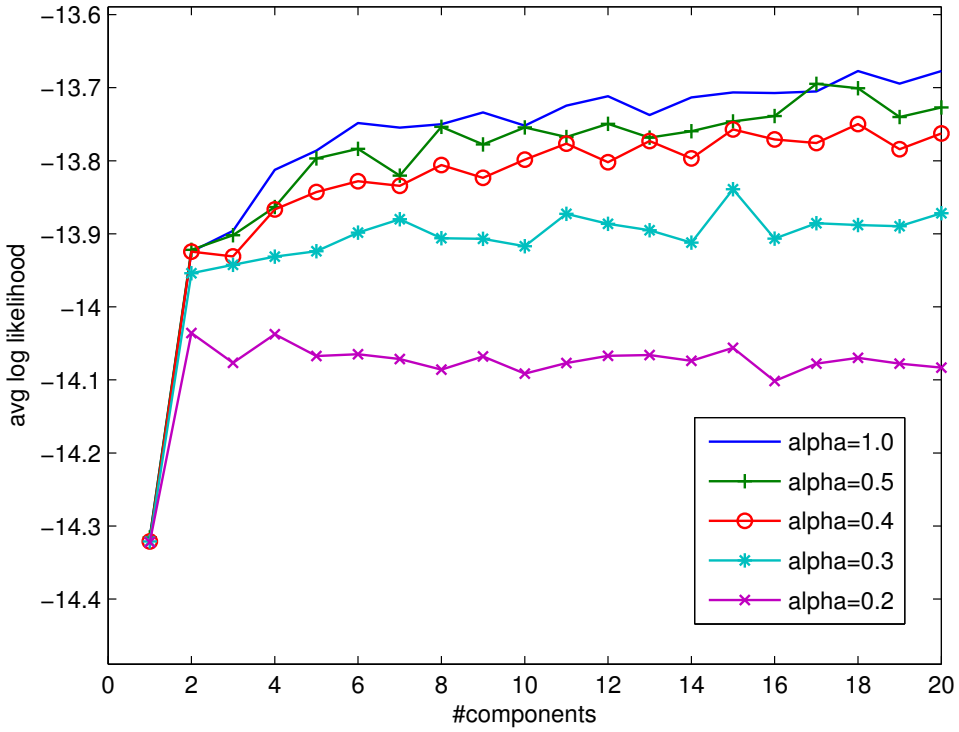

Figure 5.6: Sushi dataset. Plots of average validation log-likelihood when the training data, pairwise comparisons, are revealed with probabilities $\alpha \in \{0.2, 0.3, 0.4, 0.5, 1.0\}$. Learning degrades as α gets closer to 0.2, that is, as more pairwise comparisons are removed.

$\pi_1 = 0.17$ $\phi_1 = 0.66$	$\pi_2 = 0.15$ $\phi_2 = 0.74$	$\pi_3 = 0.17$ $\phi_3 = 0.61$	$\pi_4 = 0.18$ $\phi_4 = 0.64$	$\pi_5 = 0.16$ $\phi_5 = 0.61$	$\pi_6 = 0.18$ $\phi_6 = 0.62$
fatty tuna	shrimp	sea urchin	fatty tuna	fatty tuna	fatty tuna
salmon roe	sea eel	fatty tuna	tuna	sea urchin	sea urchin
tuna	squid	sea eel	shrimp	tuna	salmon roe
sea eel	egg	salmon roe	tuna roll	salmon roe	shrimp
tuna roll	fatty tuna	shrimp	squid	sea eel	tuna
shrimp	tuna	tuna	sea eel	tuna roll	squid
egg	tuna roll	squid	egg	shrimp	tuna roll
squid	cucumber roll	tuna roll	cucumber roll	squid	sea eel
cucumber roll	salmon roe	egg	salmon roe	egg	egg
sea urchin	sea urchin	cucumber roll	sea urchin	cucumber roll	cucumber roll

Table 5.2: Learned model for $K = 6$ (i.e., 6 mixture components) on the sushi dataset with complete preferences; π_i is the mixture proportion of the ith component while ϕ_i is the dispersion of the Mallows model in the ith component.

5.6.3.3 Movielens Data

We apply our EM algorithm to a subset of the Movielens dataset (see `www.grouplens.org`) to find "preference types" across users. We use the 200 (out of roughly 3900) most frequently rated movies, and the ratings of the 5980 users (out of roughly 6000) who rated at least one of these. Integer ratings from 1 to 5 are converted to pairwise preferences in the obvious way (for ties, no preference was added to v). For example, if A and B had rating 5, C had rating 3 and D rating 1 then the user preference becomes $v = \{A \succ C, A \succ D, B \succ C, B \succ D, C \succ D\}$. We discard preferences that are empty when restricted to the top 200 movies, and use 3986 preferences for training and 1994 for validation. We run EM with number of components $K = 1, \ldots, 20$; for each K we ran EM 20 times to mitigate the impact of local maxima. For each K, we evaluate average log-likelihood of the best run on the validation set to select the number of mixture components K. Log-likelihoods are approximated using our Monte Carlo estimator (with $K \cdot T = 120$).[3]

Log-likelihood results are shown in Figure 5.7 as a function of the number of mixture components. These results suggest that the best component sizes are $K = 10$ and $K = 5$ on the validation set. The learned model with $K = 5$ is displayed in Table 5.3, with each component ranking truncated to the top-20 movies. The five references rankings in this case are have some intuitive interpretation, but do not seem to exhibit the same separation as in the Sushi data set, in part due to the non-trivial overlap involving a number of "universally popular" movies (e.g., two movies, *The Shawshank Redemption* and *The Usual Suspects*, occur in all five components; two more occur in four, and more than 30 occur in three). Note also that the dispersion of each component is extremely high, approaching 1.

Despite this, certain patterns can be discerned. especially by focusing on reasonably *unique* movies, those that occur in only one or two components. For example, the second component contains the following "unique" movies: *Monty Python and the Holy Grail*, *The Maltese Falcon*, *Blade Runner*, *One Flew Over the Cuckoo's Nest*, *A Clockwork Orange*, *2001: A Space Odyssey*, *North by Northwest*, *Pulp Fiction*, *Chinatown*, and *Apocalypse Now*. Themes within this cluster of unique movies include "older" science fiction, ultra-violence, actor Jack Nicholson and director Stanley Kubrick. The average date of the (top) twenty movies within this component is 1970, which is significantly lower than those of other components.

[3]The C++ implementation of our algorithms have EM wall clock times of 15–20 minutes (Intel Xeon dual-core, 3GHz), certainly practical for a data set of this size. In other data sets, given the smaller number of alternatives, run times are much faster.

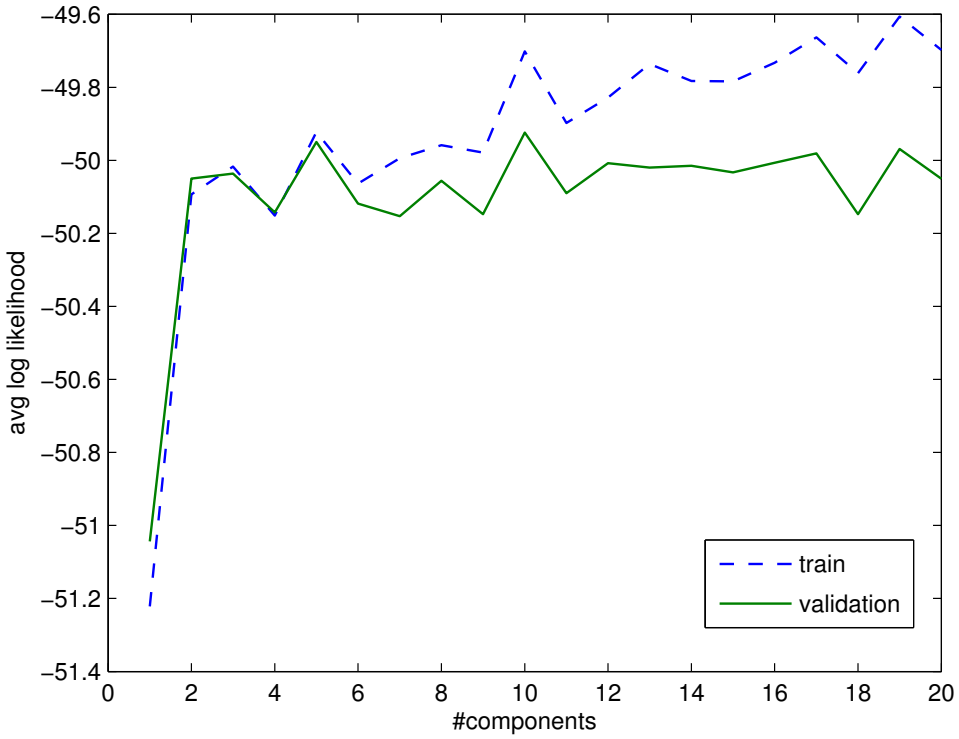

Figure 5.7: Movielens dataset: average training and validation log likelihoods on the learned model parameters of different component sizes.

The same analysis of the fifth component shows the following "unique" movies: *A Christmas Story*, *This is Spinal Tap*, *American Beauty*, *Pulp Fiction*, *The Princess Bride*, *Forrest Gump*, *Fight Club*, *Fargo*, *Ferris Bueller's Day Off*, *Raising Arizona*, *Good Will Hunting*, and *The Matrix*. Many of the movies here would commonly be characterized as "quirky," including five "quirky comedies," and several that tend toward extreme violence. The movies in this component also have a significantly later average date, 1992, than the others.

5.6.4 Predicting Missing Pairwise Preferences

In our prediction experiments, we seek to evaluate the performance of the learned models in predicting unseen pairwise comparisons. In particular, we use the complete sushi dataset, train our mixture model on the first 3500 complete rankings (we train for all $K = 1, \ldots, 20$), and select the best K by evaluating the log-likelihood on the validation dataset, which consists of 500 complete rankings. It turns out that a mixture model with $K = 6$ was most suitable.

To test posterior prediction performance, we use 1000 complete rankings, distinct

$\pi_1 = 0.24$ $\phi_1 = 0.98$	$\pi_2 = 0.23$ $\phi_2 = 0.98$	$\pi_3 = 0.21$ $\phi_3 = 0.98$	$\pi_4 = 0.19$ $\phi_4 = 0.98$	$\pi_5 = 0.13$ $\phi_5 = 0.97$
Citizen Kane (1941)	Godfather, The (1972)	Raiders of the Lost Ark (1981)	Shawshank Redemption, The (1994)	Usual Suspects, The (1995)
Godfather, The (1972)	Dr. Strangelove (1963)	Godfather, The (1972)	Life Is Beautiful (1997)	Shawshank Redemption, The (1994)
Dr. Strangelove (1963)	Citizen Kane (1941)	Schindler's List (1993)	Raiders of the Lost Ark (1981)	Schindler's List (1993)
Schindler's List (1993)	Casablanca (1942)	Rear Window (1954)	Schindler's List (1993)	Life Is Beautiful (1997)
Rear Window (1954)	Star Wars: Episode IV - A New Hope (1977)	Star Wars: Episode IV - A New Hope (1977)	Star Wars: Episode IV - A New Hope (1977)	Christmas Story, A (1983)
Shawshank Redemption, The (1994)	Usual Suspects, The (1995)	Shawshank Redemption, The (1994)	Matrix, The (1999)	This Is Spinal Tap (1984)
American Beauty (1999)	Raiders of the Lost Ark (1981)	Casablanca (1942)	Sixth Sense, The (1999)	American Beauty (1999)
Godfather: Part II, The (1974)	Monty Python and the Holy Grail (1974)	Sixth Sense, The (1999)	Sting, The (1973)	Sixth Sense, The (1999)
One Flew Over the Cuckoo's Nest (1975)	Rear Window (1954)	Psycho (1960)	Forrest Gump (1994)	Pulp Fiction (1994)
Casablanca (1942)	Maltese Falcon, The (1941)	Citizen Kane (1941)	Usual Suspects, The (1995)	Princess Bride, The (1987)
Usual Suspects, The (1995)	Blade Runner (1982)	Sting, The (1973)	Braveheart (1995)	Silence of the Lambs, The (1991)
Pulp Fiction (1994)	One Flew Over the Cuckoo's Nest (1975)	Usual Suspects, The (1995)	Green Mile, The (1999)	Godfather, The (1972)
Monty Python and the Holy Grail (1974)	Clockwork Orange, A (1971)	Saving Private Ryan (1998)	Indiana Jones and the Last Crusade (1989)	Forrest Gump (1994)
Fargo (1996)	2001: A Space Odyssey (1968)	Godfather: Part II, The (1974)	Saving Private Ryan (1998)	Fight Club (1999)
Life Is Beautiful (1997)	North by Northwest (1959)	Silence of the Lambs, The (1991)	Princess Bride, The (1987)	Fargo (1996)
Graduate, The (1967)	Pulp Fiction (1994)	Wizard of Oz, The (1939)	Star Wars: Episode V - The Empire Strikes Back (1980)	Ferris Bueller's Day Off (1986)
North by Northwest (1959)	Godfather: Part II, The (1974)	Dr. Strangelove (1963)	Silence of the Lambs, The (1991)	Raising Arizona (1987)
GoodFellas (1990)	Chinatown (1974)	Jaws (1975)	Good Will Hunting (1997)	Saving Private Ryan (1998)
Chinatown (1974)	Apocalypse Now (1979)	Braveheart (1995)	Ferris Bueller's Day Off (1986)	Good Will Hunting (1997)
Raiders of the Lost Ark (1981)	Shawshank Redemption, The (1994)	Aliens (1986)	When Harry Met Sally (1989)	Matrix, The (1999)

Table 5.3: Learned model for $K = 5$ (i.e., 5 mixture components) on Movielens; shows top 20 out of 200 movies. Parameter π_i is the mixture proportion of the ith component while ϕ_i is the dispersion of the Mallows model in the ith component.

from both the training and validation sets, and randomly remove a fraction $1 - \alpha$ of the pairwise comparisons from each ranking, then compute the transitive closure of the remaining comparisons to obtain partial preferences. We generate preferences for four different values of α. With $\alpha = 0$, all preferences are removed; with $\alpha = 0.25$, 42% of the pairwise comparisons are left after computing transitive closures; with $\alpha = 0.5$, 76% of the all pairwise comparisons remain; and with $\alpha = 0.75$, 83% of the pairwise comparisons are left.

We conditioned the learned model on the partial preferences of each agent in turn, to obtain posterior distributions over which we can infer each agent's missing pairwise comparisons. In making predictions, we use our posterior sampling algorithm SP to sample complete rankings, which we then use to update a tournament graph—recall, this is a set of counts c_{ab} to count the number of rankings for which $a \succ b$, for all $a, b \in A$. Then we estimate the posterior probability $P(a \succ b \mid v)$ by $\frac{c_{ab}}{c_{ab}+c_{ba}}$.

We define our prediction loss as follows. Suppose we have a complete ranking r with its corresponding partial preference v obtained as described above. For a given $a \succ_r b$ that is unobserved in $\mathsf{tc}(v)$, we define the posterior prediction loss to be $\hat{P}(a \prec b \mid v) = \frac{c_{ba}}{c_{ab}+c_{ba}}$. Let $M(v) = \{(a, b) : a \succ_r b, a \succ b \notin \mathsf{tc}(v)\}$ be the set of missing pairwise comparisons in v. We define the *average loss of* v as:

$$\varepsilon_v = \frac{\sum_{(a,b)\in M(v_\ell)} \hat{P}(a \prec b|v)}{|M(v_\ell)|}.$$

We next define the *average loss per preference* to be:

$$\overline{\varepsilon} = \frac{1}{n} \sum_{\ell=1}^{n} \varepsilon_{v_\ell},$$

where n is the number of distinct agents or preferences (in this case $n = 1000$). For $a \succ_r b$, let $\mathcal{D}(a, b) = r(b) - r(a)$ be the difference in their rank positions and $M_D(v) = \{(a, b) \in M(v) : \mathcal{D}(a, b) = D\}$. We also measure the *average loss at distance* D as follows:

$$\overline{\varepsilon}_D = \frac{\sum_{\ell=1}^{n} \sum_{(a,b)\in M_D(v_\ell)} \hat{P}(a \prec b|v)}{\sum_{\ell=1}^{n} |M_D(v_\ell)|}.$$

The results for average loss per preference are as follows:

- $\overline{\varepsilon} = 0.43$ for preferences generated with $\alpha = 0$;
- $\overline{\varepsilon} = 0.35$ for $\alpha = 0.25$,
- $\overline{\varepsilon} = 0.39$ for $\alpha = 0.5$, and

	$D=1$	$D=2$	$D=3$	$D=4$	$D=5$	$D=6$	$D=7$	$D=8$	$D=9$
$\alpha=0$	9000	8000	7000	6000	5000	4000	3000	2000	1000
$\alpha=.25$	6610	5487	4393	3459	2492	1744	1092	606	256
$\alpha=.5$	4429	2911	1683	898	443	191	65	22	1
$\alpha=.75$	2230	824	244	62	9	2	0	0	0

Table 5.4: The number of missing pairwise comparisons (over all agents) among pairs that are distance D from each other, with preferences generated by randomly deleting fraction $1-\alpha$ of preferences, then taking the transitive closure of the remaining comparisons.

- $\overline{\varepsilon}=0.44$ for $\alpha=0.75$.

(We interpret these results below). Results for $\overline{\varepsilon}_D$ at various distances are plotted in Figure 5.8. Since these results are extremely sensitive to the number of pairwise comparisons available in the data at different distances, we show the number of such comparisons, per distance, in Table 5.4.

The results indicate that predictive performance is weakly accurate when pairs are close in distance, but improves as the distance between the predicted alternatives increases in the underlying ranking. For $\alpha=0.75$, the average loss at distances 5 and 6 is higher than expected, but this is due to the small number of comparisons missing available for testing (and in general) at those distances. We also observe that the number of comparisons of a particular distance decreases as a function of the distance—this is more pronounced for smaller values of α. This can be attributed to the use of transitive closure: the further apart a pair of alternatives are in the underlying ranking, the less likely it is that we will remove all of the pairwise comparisons required render the two alternatives incomparable *after* taking the transitive closure of the preferences that remain. As a consequence of the skewed distribution of missing pairs available for prediction at specific distances, the average loss per preference does not in fact decrease as α increases. For example, it is 0.39 for $\alpha=0.5$, and 0.44 for $\alpha=0.75$; this is because the *relative* number of missing comparisons at smaller distances (which are more difficult to predict) is much greater when $\alpha=0.75$ than when $\alpha=0.5$ (as shown in Table 5.4).

5.7 Applications to Non-Parametric Estimators

Lebanon and Mao [79] propose a non-parametric estimator for Mallows models when observations form partitioned preferences. This estimator is an analogue of typical kernel density estimators, but over the space of rankings. Their purpose, similar to mixture models, is to model the distribution of real ranking data. The idea is to place "smooth unimodal bumps," formulated as a single Mallows model, at every input (training) pref-

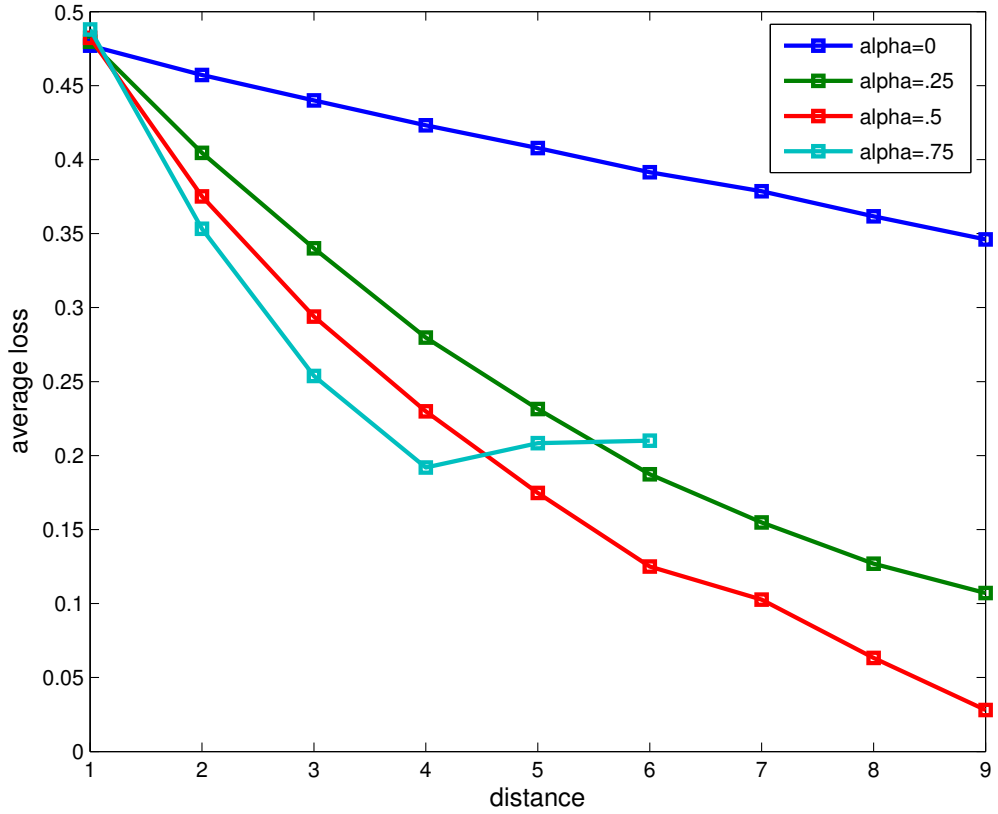

Figure 5.8: Sushi prediction results: average prediction loss for missing pairwise comparisons for pairs at different distances in the underlying ranking.

erence. This is much like a mixture model with the number of components equal to the number of preferences in the training data. They offer closed-form solutions by exploiting the existence of the closed-form for the Mallows normalization constant when partitioned preferences are observed. Unfortunately, with general pairwise comparisons, computing this normalization constant is intractable unless #P=P. In contrast to our contributions above, they do not address the question of how to find a maximum likelihood estimate of the Mallows dispersion parameter, also known as the kernel width, which they suggest as being "extremely difficult."

It turns out we can use AMP for approximate marginalization to support non-parametric estimation with general preference data. This shows the potential applicability of our sampling algorithm to a wider range of problems where observations consist of pairwise comparisons. We illustrate its application by defining a non-parametric estimator and deriving a Monte Carlo evaluation formula suitable for incomplete preferences.

Define a joint distribution q_ℓ over the probability space $\mathsf{C}(v_\ell) \times \mathsf{C}$,

$$q_\ell(s, r) = \frac{\phi^{d(r,s)}}{|\mathsf{C}(v_\ell)| Z_\phi}, \tag{5.33}$$

where Z_ϕ is the Mallows normalization constant with respect to dispersion ϕ. This distribution corresponds to drawing a ranking s uniformly at random from $\mathsf{C}(v_\ell)$ and then drawing r from a Mallows distribution with reference ranking s and dispersion ϕ. The estimator, extended in the style of Lebanon and Mao [79] to any set of paired comparisons, is simply:

$$\begin{aligned} p(v) &= \frac{1}{n} \sum_{\ell \in N} q_\ell(s \in \mathsf{C}(v_\ell), r \in \mathsf{C}(v)) \\ &= \frac{1}{n} \sum_{\ell \in N} \sum_{s \in \mathsf{C}(v_\ell)} \sum_{r \in \mathsf{C}(v)} \frac{\phi^{d(r,s)}}{|\mathsf{C}(v_\ell)| Z_\phi}. \end{aligned} \tag{5.34}$$

Note that this is a distribution over *rankings* and not incomplete preferences, that is, a marginal over $\mathsf{C}(v)$. A special case arises when $\mathbf{v}$ consists entirely of complete rankings, which simplifies to a mixture of Mallows models with n equally weighted components, each with one of the observed rankings v_ℓ as its reference ranking, and dispersion ϕ. This estimator can be useful for inference over the posterior $p(r|v) = p(r)\mathbf{1}[r \in \mathsf{C}(v)]/p(v)$ for

$r \in \mathsf{C}(v)$. For any fixed v, let $f(s) = \sum_{r \in \mathsf{C}(v)} \phi^{d(r,s)}$. Then we have:

$$\begin{aligned} p(v) &= \frac{1}{nZ_\phi} \sum_{\ell \in N} \sum_{s \in \mathsf{C}(v_\ell)} \frac{1}{|\mathsf{C}(v_\ell)|} f(s) \\ &= \frac{1}{nZ_\phi} \sum_{\ell \in N} \mathop{\mathbb{E}}_{s \sim \mathsf{C}(v_\ell)} f(s), \end{aligned}$$

where s is drawn uniformly from $\mathsf{C}(v_\ell)$. One can estimate the expectation by importance sampling. Suppose we draw, for each ℓ, rankings $s_\ell^{(1)}, \ldots, s_\ell^{(T)}$ using $\mathsf{AMP}(v_\ell, \sigma, \phi = 1)$ to approximate uniform sampling (e.g., choose some ranking σ from $\mathsf{C}(v_\ell)$). Let $w_{\ell t} = 1/\hat{P}_{v_\ell}(s_\ell^{(t)})$, which has a closed-form given by Eq. 5.15. Then the estimate is

$$\hat{p}(v) = \frac{1}{nZ_\phi} \sum_{\ell=1}^{n} \frac{\sum_{t=1}^{T} w_{\ell t} f(s_\ell^{(t)})}{\sum_{t=1}^{T} w_{\ell t}}.$$

Evaluating $f(s_\ell^{(t)})$ is generally intractable, but again, it can be approximated using our earlier techniques, as given by Eq. 5.27. In summary, we can realize non-parametric estimation using a nested sampling procedure to first approximate the outer expectation over s, followed by the inner summation $f(s)$.

5.8 Conclusion

We have developed a set of algorithms to support the efficient and effective learning of ranking or preference distributions when the observed data comprise a set of unrestricted pairwise comparisons of alternatives. Given the fundamental nature of pairwise comparisons in revealed preference, our methods extend the reach of rank learning in a vital way. One of our main technical contributions, the GRIM algorithm, allows sampling of arbitrary distributions, including Mallows models conditioned on pairwise data. It supports a tractable approximation to the #P-hard problem of log-likelihood evaluation of Mallows mixtures; and it forms the heart of an EM algorithm that was shown to be quite effective in our experiments. GRIM can also be used for non-parametric estimation.

There are a few interesting directions for future work, including various extensions and applications of the model we have developed here. One of the weaknesses with Mallows is its lack of flexibility in various dimensions, such as allowing different dispersion "rates" in different regions of the ranking. Models that allow more flexibility while controlling for overfitting could lead to more realistic ranking models for real-world settings. Other extensions include exploration of other probabilistic models of incomplete preferences that

employ different distributions over rankings, such as Plackett-Luce or weighted Mallows; that account for noisy comparison data from users; and that account for data that is not missing at random—this may occur, say, in settings in which a bias exists towards observing preferences for higher ranked alternatives.

It would also be interesting to exploit learned preference models of the type developed here for decision-theoretic tasks in social choice or personalized recommendation. Learned preferences can be leveraged in both passive (purely observational) settings and active *preference elicitation* in the context of social choice or group decision making. In Chapter 6, we will develop a probabilistic elicitation framework under which a general approach will be developed for exploiting learned preference distribution to inform and further guide the elicitation strategy. Because our algorithms allow for inference under conditional evidence, we can incrementally elicit preferences while updating a posterior preference distribution, which can be used in turn to support more effective elicitation.

Chapter 6

Elicitation with Probabilistic Preference Distributions

In Chapters 3 and 4, we developed incremental elicitation schemes that allow winner(s) to be determined with only partial preferences. However, while such elicitation strategies minimize the amount of information elicited, they tend to require repeated rounds of interaction from participants. In this chapter, we propose a probabilistic analysis of vote elicitation that combines the advantages of incremental elicitation schemes—namely, minimizing the amount of information revealed—with those of full information schemes—single (or few) rounds of elicitation. We exploit distributional models of preferences to derive the ideal *ranking threshold* k, or number of top alternatives each voter should provide, to ensure that either a winning or a high quality alternative (as measured by max regret) can be found with high probability.

This chapter is organized as follows. In Section 6.1, we provide some background and illustrate some common pitfalls of full incremental elicitation, which will motivate our multi-round model. In Section 6.2, we introduce our multi-round, probabilistic vote elicitation model. In Section 6.3, we study single-round top-k elicitation, and we develop a general empirical methodology, which uses preference profile samples to determine the ideal ranking threshold for many common voting rules. We also develop probably approximately correct (PAC) sample complexity results for one-round protocols with *any voting rule*. In Section 6.4, we demonstrate the efficacy of our approach empirically on one-round protocols with Borda scoring.

6.1 Motivation

While our incremental elicitation protocols developed in Chapters 3 and 4 minimize the amount of preference information requested from voters in a practical sense (before a good alternative can be recommended), they suffer from their incremental nature: the choice of the next voter to query and the query to pose depends on all previous responses. This may result in interruption costs and unacceptable latency on the part of agents. If agent ℓ was just asked if a is preferred to b, ℓ must wait for the elicitor to query potentially many other agents before posing another query to ℓ. This can be less desirable than being asked a batch of pairwise comparisons in one shot, since in batch mode the agent only needs to interact with the elicitor once or a few times.

To make this concrete, consider an example of making a scheduling decision for a meeting of three people: Alice, Bob and Carol. The alternatives represent the start times of the meeting. Let's assume that the participants are remotely distributed or otherwise unable to assemble together, and that elicitation is conducted through an electronic channel such as email or instant messaging.

Incremental elicitation would proceed by asking, say Alice, if she prefers 10am over 9am. Since Alice does not check her email often, Alice responds an hour later, in favour of 10am after checking her calendar. Then Bob is queried next, being asked to choose 11am over 10am. Bob is relatively fast and answers within fifteen minutes, preferring 10am. Then elicitation proceeds to Carol, asking her if she prefers 10am over 1pm. Carol responds in half an hour and prefers 10am. Elicitation then goes back to Alice, after spending a total of forty-five minutes to work through Bob and Carol. This process continues and may well take a few more hours before reaching a consensus meeting time. Clearly, this process takes much longer to reach a decision, putting a significant time cost on incremental elicitation, despite the fact that a minimal amount of information has been elicited. Furthermore, there is potentially an "interruption cost" imposed on each individual, where an individual experiences multiple distractions as new queries are posed.

On the other hand, if the elicitation process queried each person, simultaneously, by asking for their *most preferred* time slot, they would all answer 10am and a consensus decision would be reached much more quickly—in an hour if we assume the same response latency as above (i.e., the time for the slowest individual to respond). Thus, while this elicitation approach asks many more pairwise comparisons per agent (since a top-1 query implicitly consists of $m-1$ pairwise comparisons), it is overall much faster, and reduces interruptions and waiting times while still asking only a fraction of the full ranking

information.

In this chapter, we develop an approach to vote elicitation that introduces the concept of *rounds* of elicitation (e.g., asking a top-1 query to all agents simultaneously is considered to be one round). Furthermore, our approach exploits distributional information over voter preferences, that are, for example, learned using techniques from Chapter 5, to reduce *both* the amount of information elicited from voters *and* the number of rounds of elicitation. Indeed, these factors can be explicitly traded off against one another. Our model also supports approximation, using minimax regret to further minimize the amount of information elicited, the number of rounds, or both. In this way, we provide the first framework that allows the design of vote elicitation schemes that address the complex three-way tradeoff between approximation quality, total information elicited, and the number of rounds of elicitation.

Developing analytical bounds on these performance dimensions depends, of course, on the specific distributional assumptions about the preferences and the voting rule in question. While we make some suggestions regarding the types of results one might derive along these lines, our primary contribution is an empirical methodology that allows a designer to assess these tradeoffs and design elicitation schemes for *any preference distribution*, and *any voting rule* that can be interpreted using some form of scoring. To illustrate the use of both our general elicitation framework and our empirical methodology, we analyze one-round vote elicitation protocols. We develop general PAC sample complexity bounds for such one-round protocols. We then analyze these protocols empirically using Mallows models of preferences distributions [87, 88] and Borda scoring as the voting protocol. Our results suggest that good, even optimal, results can be obtained in one-round protocols even when only a small portion of the preferences of the voters is elicited.

Recently, Filmus and Oren [46] give upper and lower bounds for the value of the parameter k, in one-round top-k elicitation, that's necessary to guarantee the true winner can be determined with high probability. Their primary results pertain to positional scoring rules and three types of preference distributions. Given an arbitrary PSF, they provide a set of criteria for obtaining upper and lower bounds for k. They apply the criteria and derive bounds for the Borda, harmonic ($\alpha(i) = 1/i$, for $i \leq m$), and geometric ($\alpha(i) = z^i$ for $i \leq m$ and $z \in (0, 1)$) scoring rules. In particular, they show that with Borda any elicitor requires $k = \Omega(m)$ (one can determine the winner by querying for the entire ranking, with $k = m - 1 = O(m)$). For the harmonic PSF, they devise an algorithm that can determine the winner—with high probability—after eliciting top-k preferences with $k = \omega(\log^{4/3} m)$, and for the geometric PSF, the same algorithm can

achieve the same guarantees with only $k = \omega(\log \log m)$.

Kalech et al. [69] develop several heuristic strategies for vote elicitation, including one scheme that proceeds in rounds in which voters provide larger "chunks" of information. This offers an advantage over the fully incremental schemes presented in Chapters 3 and 4. Unfortunately, Kalech et al.'s approach does not admit approximation (with quality guarantees), and no principles are provided to select an appropriate chunk size.

The results in this chapter are primarily concerned with single-winner score-based voting rules. These are rules that score the "societal welfare" of an alternative from a given preference profile, and where the alternative(s) with the highest score is selected as the winner. Indeed, all voting rules can be cast as a score-based rule by assigning a score of 1 to the selected winner and zero scores to all other alternatives. However, many common and popular scoring rules admit a natural scoring function.

Our experiments in this chapter uses the Borda rule, which, for a given ranking, scores an alternative ranked at position i with a score of $m - i$. But our results extend more generally to other score-based voting rules. We do not address multi-winner schemes, but the same underlying principles can be readily extended to that setting. We refer the reader to Section 2.1 for a more comprehensive overview of preference relations, partial preferences, and single-winner voting rules (including Borda and other scoring rules).

Given a partial preference profile $\mathbf{p} = (p_1, \ldots, p_n)$, we will use minimax regret (see Section 3.1.1, Eqs. 3.1—3.5) as a robust decision criterion for winner determination with partial information. Note that this notion gives us a form of robustness in the face of vote uncertainty: every alternative has worst-case loss (with respect to the voting rule's score) at least as great as that of the minimax optimal alternative. Notice that if $MMR(\mathbf{p}) = 0$, then the minimax winner is in fact optimal, regardless of the completion of the partial profile. MMR can be computed in polynomial time for several common voting rules, including Borda.

6.2 A Model of Multi-round Probabilistic Elicitation

We begin by developing a general model of vote elicitation that allows one to make explicit tradeoffs between the number of rounds of elicitation, the amount of information provided by each voter, and approximation quality. Let a *query* refer to a "single" request for information from a voter. Types of queries include simple pairwise comparisons (e.g., "Do you prefer a to b?"); sets of such comparisons; more involved partial requests (e.g., "Who are your top k candidates?"); or requests for entire rankings. Different queries have different "costs"—both in terms of voter cognitive effort and communication costs (which

range from 1 to roughly $m \log m$ bits)—and provide varying degrees of information.

Given a particular class of queries $\mathcal{Q}$, informally, a *multi-round voting protocol* selects, at each round, a subset of voters, and one query per selected voter. The *voter-query (VQ) pairs* selected at round t can be conditioned on the responses to all previous queries. More formally, let I_{t-1} be the *information set* available at round $t \geq 1$ (i.e., responses to queries at rounds $1, \ldots, t-1$). We assume, in this chapter, that all queries used admit responses that can be represented as a set of pairwise comparisons. Let $\mathbf{P}$ denote the set of all partial preference profiles with respect to a fixed n and m. As such, this information set I_{t-1} can be represented as a sequence of t partial profiles $(\mathbf{p}^0, \ldots, \mathbf{p}^t) \in \mathbf{P}^t$.

Definition 22. A *multi-round voting protocol* is a tuple $\Pi = \langle \mathcal{Q}, \boldsymbol{\pi}, \omega \rangle$, where

(a) $\mathcal{Q}$ is a *class of preference queries*,

(b) $\boldsymbol{\pi} = (\pi^t)_{t \geq 1}$ is a sequence of *querying functions* such that for all $t \geq 1$, $\pi^t : \mathbf{P}^t \to (N \to \mathcal{Q} \cup \{\perp\})$, and

(c) $\omega : \mathbf{P} \to A \cup \{\perp\}$ is a *winner selection function*.

If $\omega(\mathbf{p}^t) = \perp$, no winner is declared at round t and the protocol proceeds to round $t+1$. Otherwise the protocol terminates with the chosen winner at round t. If $\pi^t(\mathbf{p}^{t-1})(\ell) = \perp$, then no query is posed to agent ℓ at round t.

Definition 23. Let $\Pi = \langle \mathcal{Q}, \boldsymbol{\pi}, \omega \rangle$ be a multi-round voting protocol, and f a social choice function. The protocol Π is an *exact protocol* for f if, for all preference profiles $\mathbf{v}$, starting with an empty partial profile $\mathbf{p}^0$ (i.e., $\mathbf{p}^0 = \emptyset^n$), each round of responses $\mathbf{p}^{t-1}$ is consistent with $\mathbf{v}$, and there exists a round t^* such that $\omega(\mathbf{p}^{t^*-1}) = f(\mathbf{v})$.

In other words, Π is an exact protocol if it continues eliciting preferences until it has enough preference information to determine the true winner under SCF f.

Suppose we have a distribution P over full preference profiles. Given a protocol Π, we have an induced distribution over runs of Π (with respect to responses being consistent with a profile randomly drawn from P at the beginning of the elicitation process), which in turn gives us a distribution over various properties reflecting the cost and performance of Π. There are three general properties of interest to us:

(a) *Quality of the winner*: if Π terminates with information set $\mathbf{p}^{t^*-1}$ and winner $a = \omega(\mathbf{p}^{t^*-1})$, we can measure quality using either *expected regret*,

$$\sum_{\mathbf{v}} Regret(a, \mathbf{v}) P(\mathbf{v} | \mathbf{p}^{t^*-1}),$$

or *maximum regret*,

$$MR(a, \mathbf{p}^{t^*-1}).$$

If Π is an exact protocol, both measures will be zero. We focus here on max regret, which provides worst-case guarantees on winner quality. In some settings, expected regret might be more suitable.

(b) *Amount of information elicited*: this can be measured in various ways (e.g., equivalent number of pairwise comparisons or bits).

(c) *Number of rounds* of elicitation.

There is a clear tradeoff between these factors. A greater degree of approximation in winner selection can be used to reduce informational requirements, rounds, or both [84]. For any fixed quality threshold, the number of rounds and the amount of information elicited can also be traded off against one another. At one extreme, optimal outcomes can clearly be found in one round if we ask each voter for full rankings. At the other extreme, optimal policies minimizing expected elicited information can always be constructed (though this will likely come at great computational expense) by selecting a single VQ-pair at each round, where each query carries very little information (e.g., a simple pairwise comparison), at a dramatic cost in terms of number of rounds. How one addresses these tradeoffs depends on the costs associated with each of these factors. For example, the cost of elicited information might reflect the number and type of queries asked of voters, while the cost associated with rounds might reflect interruption and delay experienced by voters as they "wait" for other voters to answer queries before receiving their own next query.[1]

Computing optimal protocols for specific voting rules, query classes, distributions over preferences, and cost models is a very important problem that can be addressed explicitly using our framework. The framework supports both Bayesian and PAC-style (probably approximately correct) analysis. We illustrate its use by considering a specific type of protocol using a PAC-style analysis in the next section.

[1]We're being somewhat informal, since some voters may only be queried at subset of the rounds. If a (conditional) sequence of queries is asked of a single voter ℓ without any interleaving queries to another voter j, we might count this as a single "session" or round for ℓ. These distinctions won't be important in what follows.

6.3 Probably Approximately Correct One-round Protocols

Imagine we require a *one-round protocol*, where each voter can be asked, exactly once, to list their top-k alternatives. A natural question is: what is the minimum value k^* for which such top-k queries ensure that the resulting partial profile $\mathbf{p}$ has low minimax regret, $MMR(\mathbf{p}) \leq \varepsilon$, with high probability, say at least $1-\delta$? We call ε and δ the *minimax regret accuracy* and *confidence* parameters, respectively. Obviously, such a k^* exists: with $k = m-1$, we elicit each voter's full ranking, always ensuring $MMR(\mathbf{p}) = 0$. This question is of interest when, for example, more than one round of elicitation is infeasible or very costly, an approximate solution (with tolerance ε) is suitable, and some small probability δ of a poor solution is acceptable.

Let $\mathbf{p}[k]$ denote the restriction of full preference profile $\mathbf{v} = (v_1, \ldots, v_n)$ to the sub-rankings consisting of each voter's top-k alternatives. For any distribution P over voter preferences $\mathbf{v}$, $MMR(\mathbf{p}[k])$ is a random variable. Let $q_k = P(MMR(\mathbf{p}[k]) \leq \varepsilon)$. We would like to find $k^* = \min\{k : q_k \geq 1-\delta\}$. Even if we assume P has a particular form, computing k^* might be analytically intractable; or the analytically derived upper bounds may too loose to be of practical use. If one can instead sample vote profiles from the true distribution—without necessarily knowing what P is—a simple empirical methodology can be used to determine a small $\hat{k}$ that, with high probability, has the desired MMR accuracy with near the desired MMR confidence (see Theorem 19 below). Specifically, we take the following steps:

(a) Specify the following parameters: MMR accuracy $\varepsilon > 0$, MMR confidence $\delta > 0$, *sampling accuracy* $\xi > 0$, and *sampling confidence* $\eta > 0$.

(b) Obtain t i.i.d. samples of full preference profiles $S = (\mathbf{v}_1, \ldots, \mathbf{v}_t)$ where

$$t \geq \frac{1}{2\xi^2} \ln \frac{2(m-2)}{\eta}. \tag{6.1}$$

(c) Output $\hat{k}$, the smallest k for which

$$\hat{q}_k \equiv \frac{|\{i \leq t : MMR(\mathbf{p}_i[k]) \leq \varepsilon\}|}{t} > 1 - \delta - \xi \ .$$

The parameters ξ and η are required to account for sampling randomness, and are incorporated as part of the statistical guarantee on the algorithm's success (see Theorem 19).

In summary, the approach is to estimate q_k (which is usually intractable to derive analytically) using $\hat{q}_k$, and take the smallest $\hat{k}$ that, accounting for sampling error, is highly likely to have the true probability, $q_{\hat{k}}$, lie close to the desired MMR confidence threshold $1-\delta$. The larger the sample size t, the better the estimates, resulting in smaller ξ and η. Using a sample set specified as in the algorithm, one can obtain a PAC-style guarantee [116] on the quality of one-round, top-$\hat{k}$ elicitation:

Theorem 19. *Let $\varepsilon, \delta, \eta, \xi > 0$. If the sample size t satisfies Eq. (6.1), then for any preference profile distribution P, with probability $1-\eta$ over i.i.d. samples $\mathbf{v}_1, \ldots, \mathbf{v}_t$, we have: (a) $\hat{k} \leq k^*$; and (b) $P[MMR(\mathbf{p}[\hat{k}]) \leq \varepsilon] > 1-\delta-2\xi$.*

Proof. For any $k \leq m-2$ (for $k=0$, minimax regret is $n(m-1)$ and for $k \geq m-1$ minimax regret is 0, so we are not interested in these cases), the indicator random variables $\mathbf{1}[MMR(\mathbf{p}_i[k]) \leq \epsilon]$ for $i \leq t$ are i.i.d. By the Hoeffding bound, we have

$$\Pr_{S \sim P^t}\left[|\hat{q}_k - q_k| \geq \xi\right] \leq 2\exp(-2\xi^2 t).$$

If we choose t such that $\frac{\eta}{m-2} \leq 2\exp(-2\xi^2 t)$ we obtain Inequality (6.1) and

$$\begin{aligned}
&\Pr_{S \sim P^t}\left((|\hat{q}_1 - q_1| \leq \xi) \wedge (|\hat{q}_2 - q_2| \leq \xi) \wedge \ldots \wedge (|\hat{q}_{m-2} - q_{m-2}| \leq \xi)\right) \\
&= 1 - \Pr_{S \sim P^t}\left[\bigcup_{k=1}^{m-2} |\hat{q}_k - q_k| > \xi\right] \\
&\geq 1 - (m-2)\cdot\frac{\eta}{m-2} \qquad (6.2) \\
&= 1-\eta,
\end{aligned}$$

where Inequality 6.2 follows from the union bound. Thus with probability at least $1-\eta$, uniform convergence holds, and we have $\hat{q}_{k^*} > q_{k^*} - \xi > 1-\delta-\xi$. Since $\hat{k}$ is the smallest k with $\hat{q}_k > 1-\delta-\xi$ we have $\hat{k} \leq k^*$. Furthermore, $q_{\hat{k}} > \hat{q}_{\hat{k}} - \xi > (1-\delta-\xi)-\xi = 1-\delta-2\xi$, which shows part (b). ■ □

We note several significant aspects of this result. First, it is *distribution-independent*—we need t i.i.d. samples from P, where t depends only on ξ, η and m, and *not* on any property of P. Of course, depending on the nature of the distribution, the required sample size may be larger than necessary (e.g., if P is highly concentrated).

Second, note that an algorithm that outputs $k = m-1$ guarantees $MMR = 0$, but is effectively useless to the elicitor; hence we desire an algorithm that proposes a k that is not much larger than the optimal k^*. Our scheme guarantees $\hat{k} \leq k^*$.

Third, while the true probability $q_{\hat{k}}$ of the estimated $\hat{k}$ satisfying the regret accuracy requirement may not meet the confidence threshold, it lies within some small tolerance of that threshold. This is unavoidable in general. For instance, if we have $q_{k^*} = 1 - \delta$, there is potentially a significant probability that $\hat{q}_{k^*} < 1 - \delta$ for any finite sample; but our result ensures that there is only a small probability that $\hat{q}_{k^*} < 1 - \delta - \xi$.

Fourth, part (b) of Theorem 19 remains valid if the sum $\delta + \xi$ is fixed (and in some sense, this sum can be interpreted as our ultimate confidence); but variation in δ and ξ does impact sample size (and part (a)). One can reduce the required sample size by making ξ larger and reducing δ correspondingly, maintaining the same "total" degree of confidence, but the guarantee in part (a) becomes weaker since k^* generally increases as δ decreases. This is a subtle tradeoff that should be accounted for in the design of an elicitation protocol.

We can provide no *a priori* guarantees on how small k^* might be, since this depends crucially on properties of the distribution; in fact, it might be quite large (relative to m) for, say, the impartial culture model (as we see below). But our theorem provides a guarantee on the size of $\hat{k}$ with respect to the optimal k^*.

An analogous result can easily be obtained if one is interested in determining the smallest k for a one-round protocol that has small *expected MMR*. However, using expectation does not preclude MMR from being greater than a desired threshold with significant probability. Hence, expected MMR as a decision criterion may be ill-suited to choosing k in many voting settings. The techniques above can also be used in a Bayesian fashion, where instead of using minimax regret to determine robust winners, one uses *expected regret* (i.e., expected loss relative to the optimal candidate given uncertainty over completions the partial profile).

Our empirical methodology can also be used in a more heuristic fashion, *without* derivation of precise confidence bounds. One can simply generate random profiles, use the empirical distribution over $MMR(\mathbf{p}[k])$ as an estimate of the true distribution over minimax regret, and select the desired k based directly on properties of the empirical distribution (e.g., represented as histograms, as we illustrate in the next section).

Finally, we note that samples can be obtained in a variety of ways, e.g., drawn from a learned preference model, such as a Mallows model or Mallows mixture (e.g., using RIM), or simply obtained from historical problem instances. In multi-round protocols, the GRIM model can be used to realize conditional sampling if needed. Our empirical methodology is especially attractive when k^* cannot easily be derived analytically (which may well be the case for Mallows, Plackett-Luce, and other common models). We note that while Filmus and Oren [46] derive k^* for Mallows, their results apply in an asymptotic

sense, with respect to n and m, and therefore do not provide a practical approach to the selection of k^*.

6.4 Empirical Evaluation

To explore the effectiveness of our methodology, we ran a suite of experiments, sampling voter preferences from Mallows models using a range of parameters, computing minimax regret for each sampled profile for various k, and estimating both the expected minimax regret and the MMR-distribution empirically. We also discuss experiments with two real-world data sets. Borda scoring is used in all experiments.

For the Mallows experiments, a preference profile is constructed by drawing n i.i.d. rankings, one per voter, from a fixed Mallows model. Each experiment varies the number of voters n, number of alternatives m, and dispersion ϕ, and uses 100 preference profiles. We simulate the elicitation of top-k preferences and measure both MMR and true regret (with respect to the true preferences and true winner) for $k = 1, \ldots, m-1$; results are "normalized" by reporting max regret and true regret *per voter*. Figures 6.1 and 6.2 shows histograms reflecting the empirical distribution of both MMR and true regret for various k, ϕ, n, and m. That is, in each collection of histograms, as defined by particular (m, n, ϕ) parameter values, we generated 100 instances of random preference profiles. For each instance of a profile, and each k, we compute MMR of the partial votes when top-k preferences are revealed in the profile—this represents one data point along the horizontal axis, in the histogram corresponding to that particular k, and to parameters values (m, n, ϕ). Note that (normalized) MMR per voter can range from 0 to 9 since we use Borda scoring.

Clearly MMR is always zero when $k = m - 1 = 9$. For small ϕ (e.g., $0.1, 0.4$), preferences across voters are reasonably similar, and values of $k = 1$–3 are usually sufficient to find the true winner, or at least one with small max regret (see top two plots in Figure 6.1). But even with $m = 10$, $n = 100$ and $\phi = 0.6$, $k = 4$ results in a very good approximate winner: $MMR \leq 0.6$ in 90/100 instances (see plot in second row and first column of Figure 6.1). Even the most difficult case for partial elicitation—the uniform distribution with $\phi = 1$—gives reasonable MMR guarantees with high probability with less than full elicitation ($k = 5$–7, depending on one's tolerance; see plot in second row and second column of Figure 6.1). The heuristic use of the empirical distribution in this fashion is likely to suffice in practice in a variety of settings; but we can apply the theoretical bounds above as well. Since we have $t = 100$ (admittedly a small sample), by Eq. (6.1), we can set $\eta = 0.05$ and $\xi = 0.17$, and with $\delta = 0.9$, $\varepsilon = 0.5$, we obtain $\hat{k} = 4$.

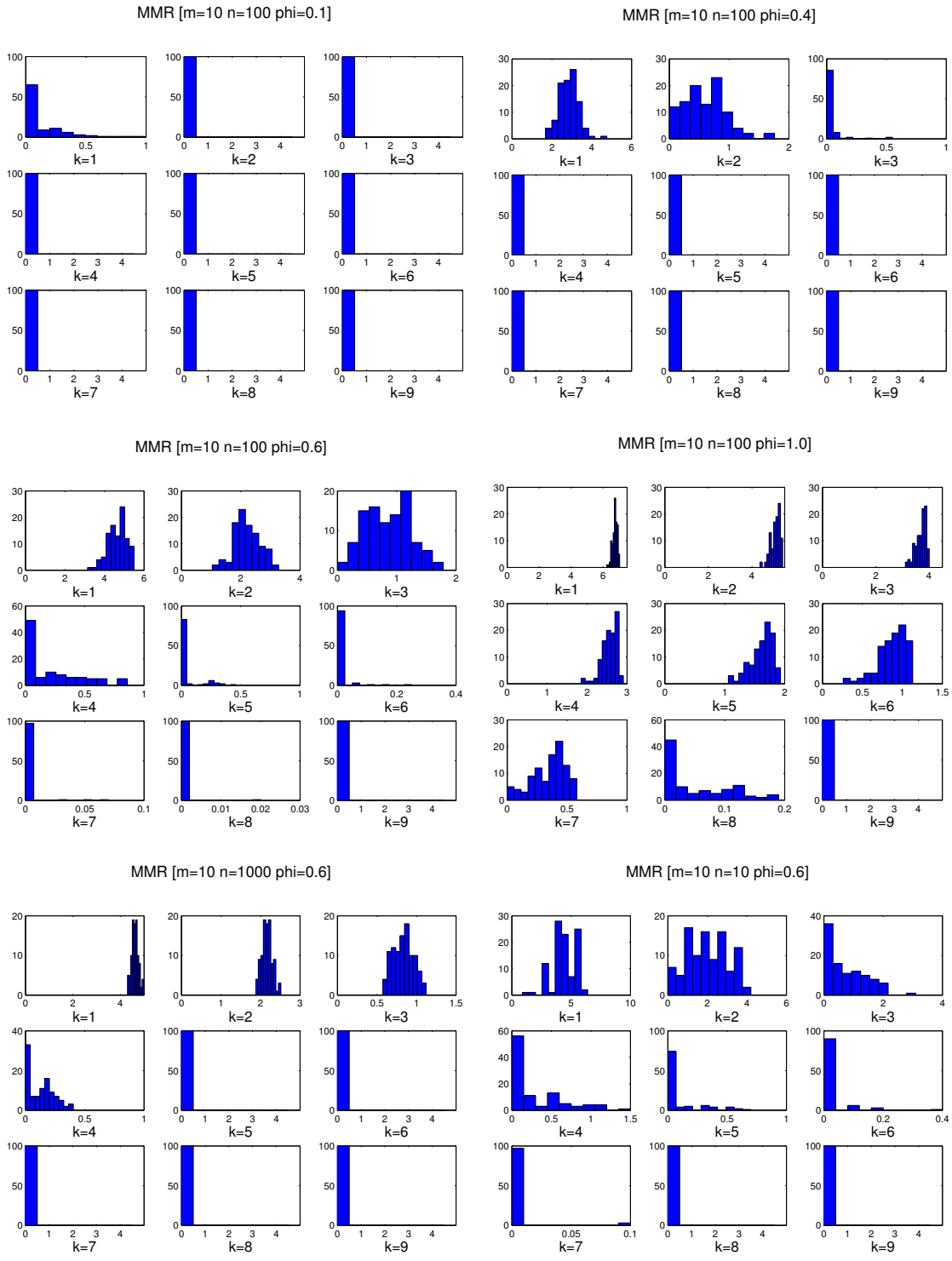

Figure 6.1: MMR plots for various ϕ, n and m: for $m = 10$, $n = 100$, and $\phi \in \{0.1, 0.4, 0.6, 1.0\}$; for $\phi = 0.6$, $m = 10$ and $n \in \{10, 1000\}$. Histograms show, after eliciting top-k, avg. MMR per voter on the x-axis and number of profile instances on the y-axis.

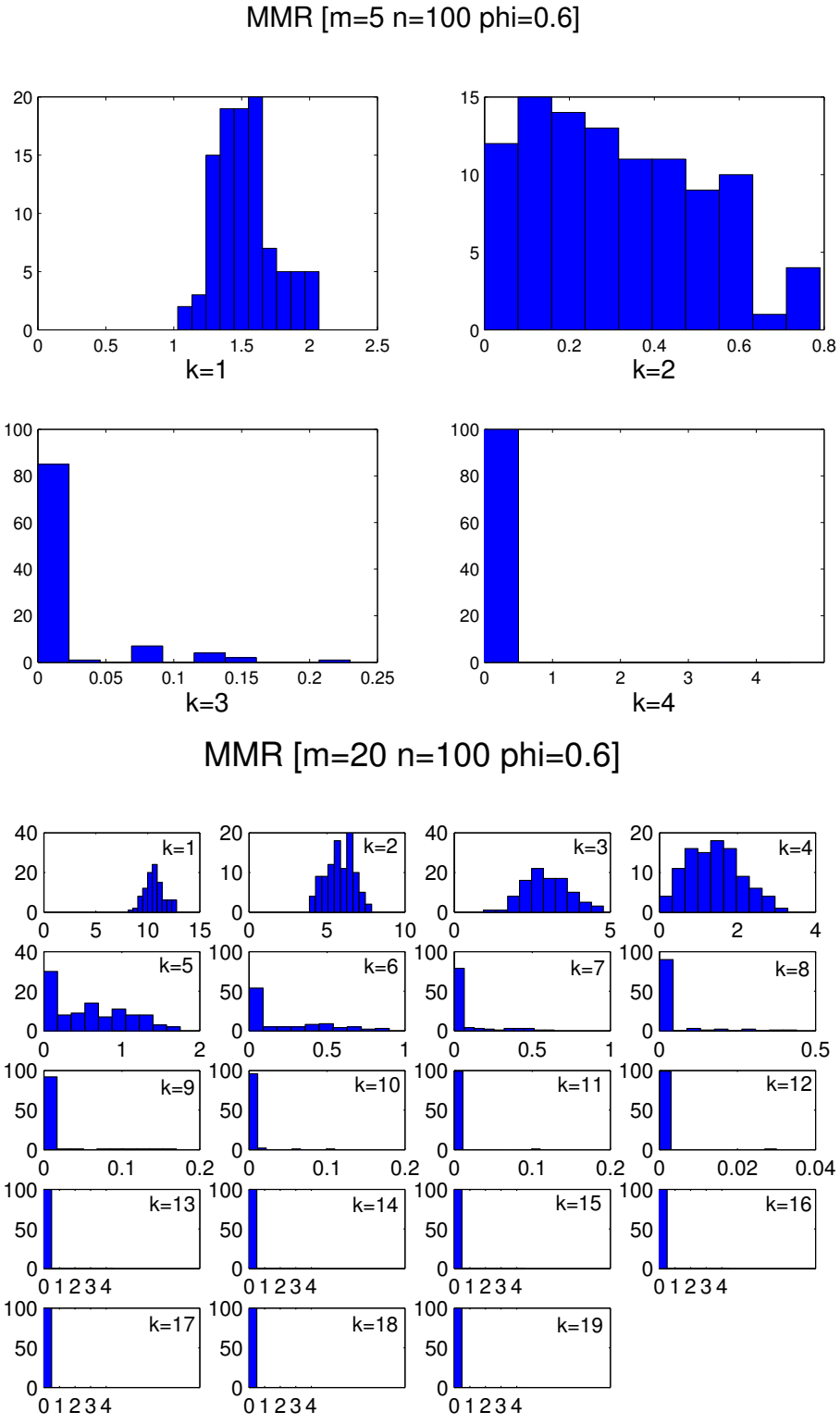

Figure 6.2: MMR plots for $m = 5$, $\phi = 0.6$; and $m = 20$, $\phi = 0.6$. Histograms show, after eliciting top-k, avg. MMR per voter on the x-axis and number of profile instances on the y-axis.

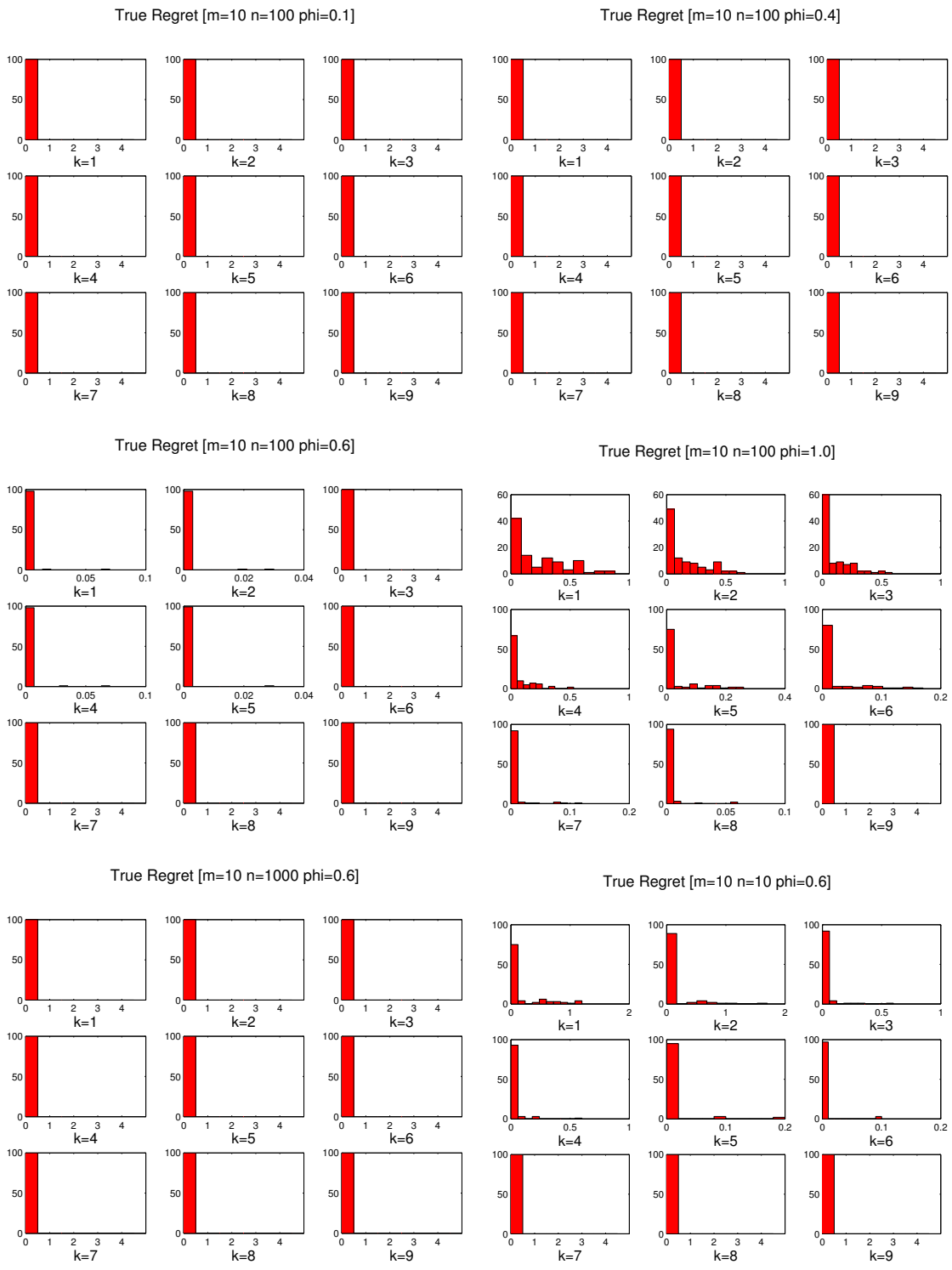

Figure 6.3: The corresponding true regrets of experiments shown in Figure 6.1. Histograms show, after eliciting top-k, avg. regret per voter on the x-axis and number of profile instances on the y-axis.

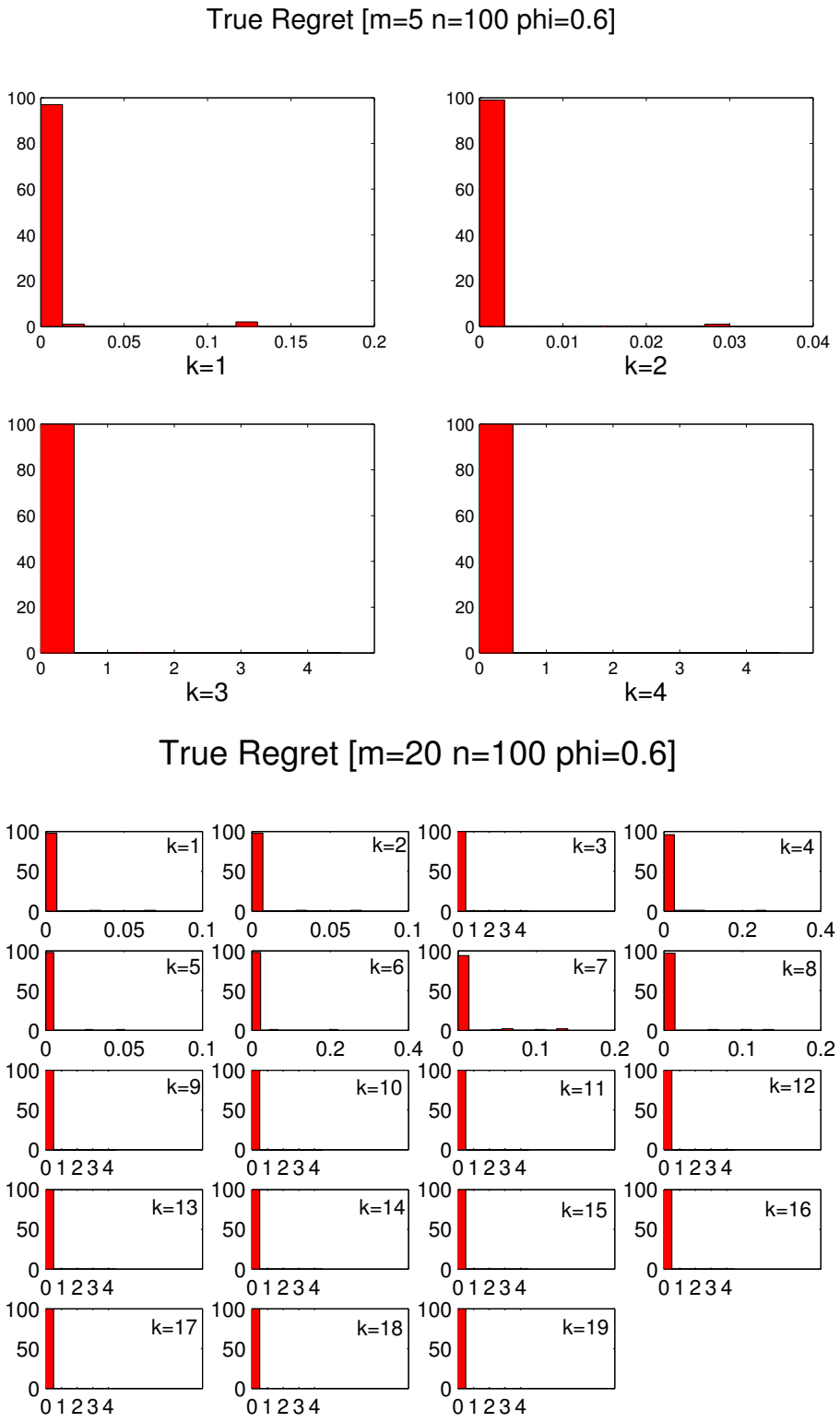

Figure 6.4: The corresponding true regrets of experiments shown in Figure 6.2. Histograms show, after eliciting top-k, avg. regret per voter on the x-axis and number of profile instances on the y-axis.

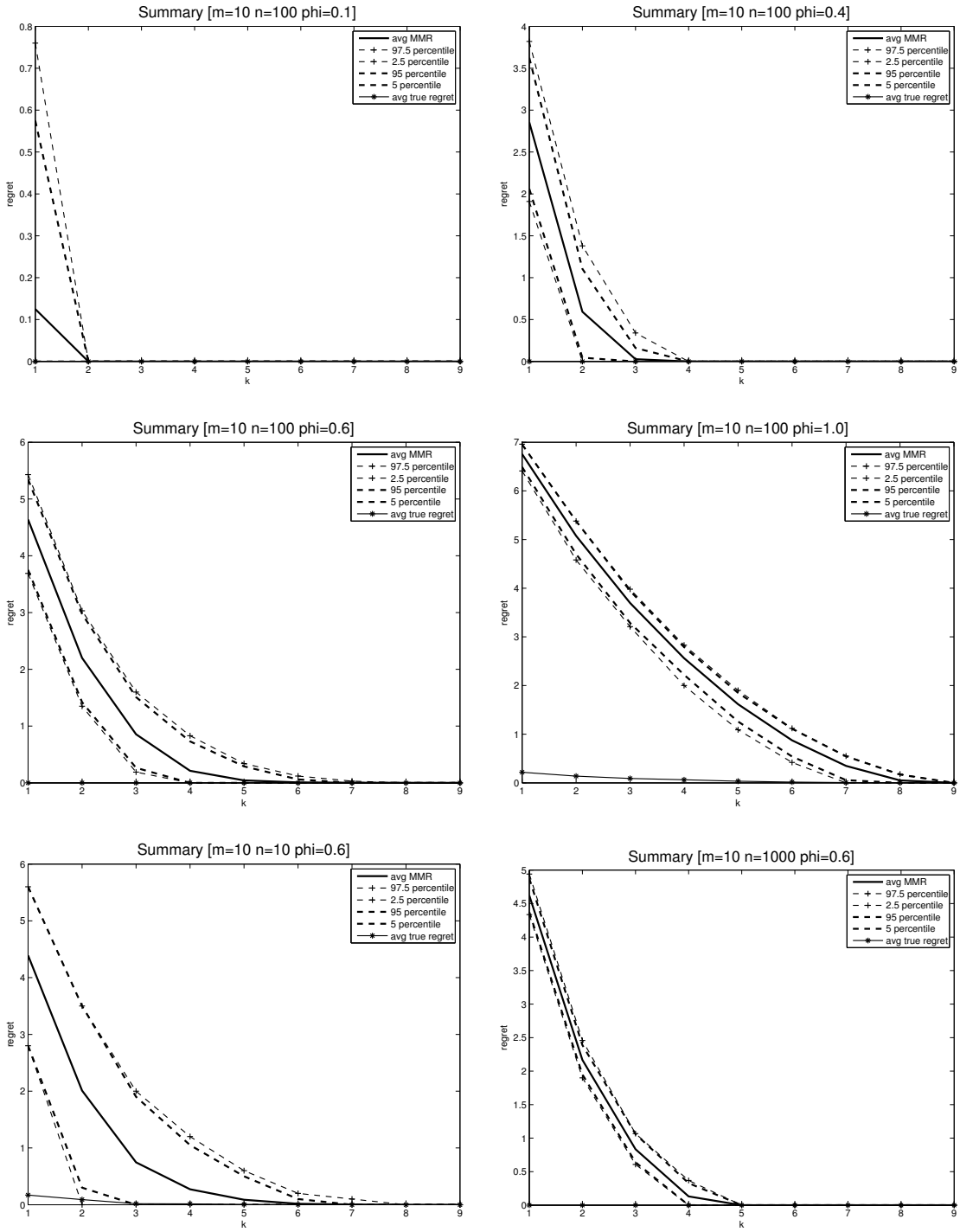

Figure 6.5: Each plot corresponds to a summary of the experiments in Figure 6.1, and shows the reduction in regret (avg. per voter, MMR and true regret over all instances) as k increases. Percentiles (2.5%, 5%, 95%, 97.5%) for MMR are shown over 100 profiles.

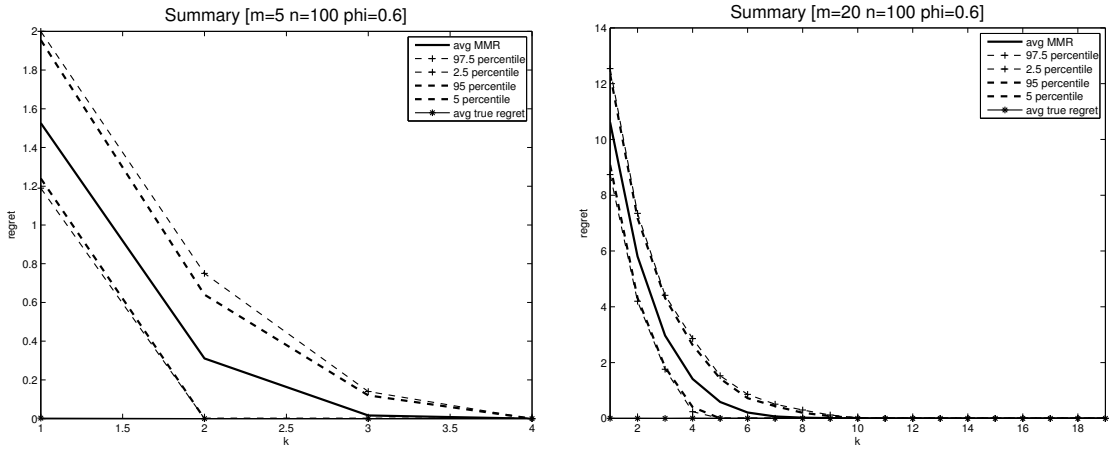

Figure 6.6: Each plot corresponds to a summary of the experiments in Figure 6.2. Plots show the reduction in regret (avg. per voter, MMR and true regret over all instances) as k increases. Percentiles (2.5%, 5%, 95%, 97.5%) for MMR are shown over 100 profiles.

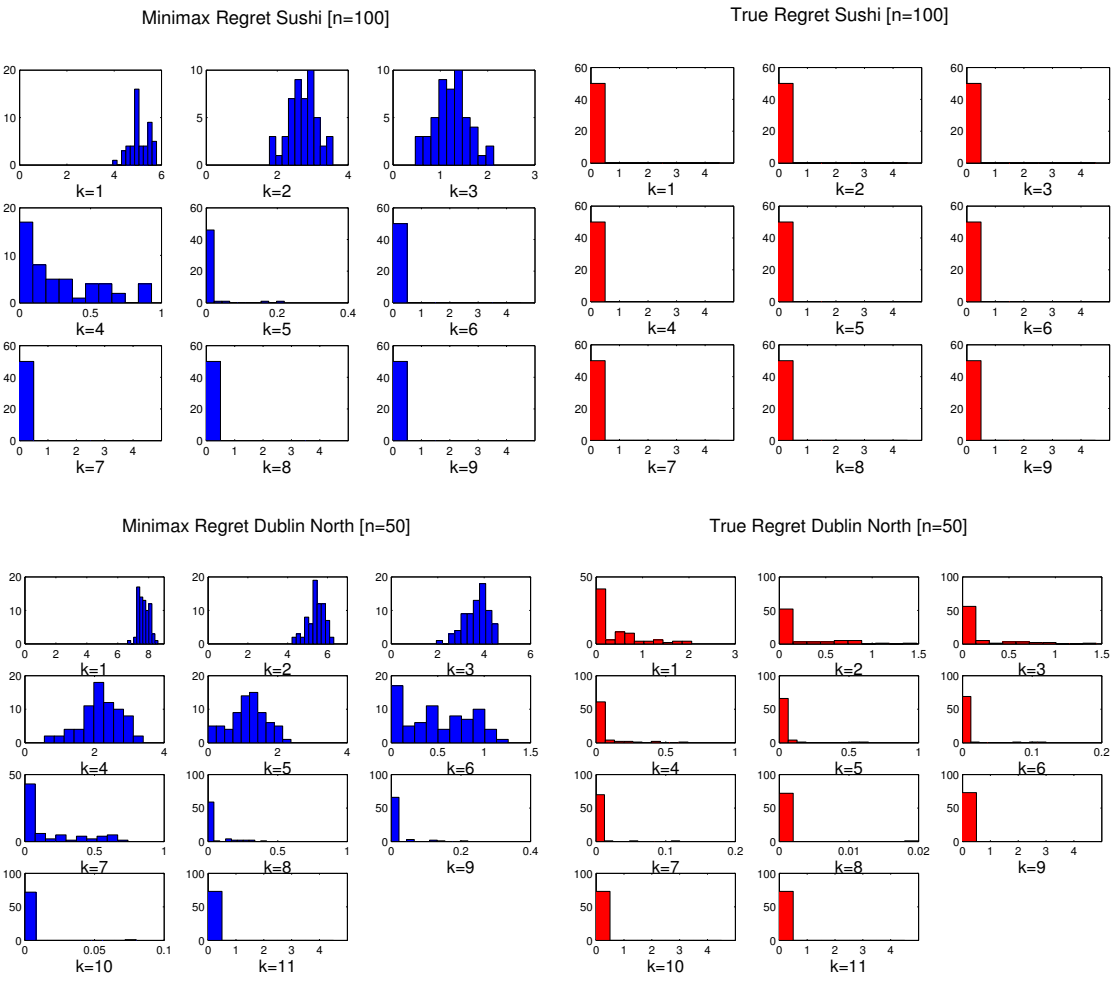

Figure 6.7: Results on Sushi and Irish. Histograms show, after eliciting top-k, avg. MMR and true regret per voter on the x-axis and number of profile instances on the y-axis.

By Theorem 19, we are guaranteed with probability 0.95 that $\hat{k} \leq k^*$ and $q_{\hat{k}} > 0.56$. If we wanted $q_{\hat{k}}$ to be closer to 0.9, then requiring $t \geq 28842$ gives $\xi = 0.01$ and $q_{\hat{k}} > 0.88$.

True regret (see Figures 6.3 and 6.4) is even more illuminating: with $\phi = 0.6$, the MMR solution *after only top*-1 *queries to each voter* is nearly always the true winner; and true regret never exceeds 2. Even for the uniform distribution with $\phi = 1$, true regret is surprisingly small: after top-2 queries, regret is less than 0.5 in 97/100 cases. As we increase the number of voters n, the MMR distribution becomes more concentrated around the mean (e.g., $n = 1000$), and often resembles a Gaussian. Roughly, this is because with Borda scoring, (normalized) MMR can be expressed as the average of independent functions of p_ℓ through pairwise max regret $PMR_\ell(a^*_{\mathbf{p}}, a') = \max_{v_\ell \in \mathbb{C}(p_\ell)} B(v_\ell(a')) - B(v_\ell(a^*_{\mathbf{p}}))$, where a' is the adversarial witness (see Eq. 3.1).

Figures 6.5 and 6.6 provides a summary of the above experiments, showing average MMR as a function of k, along with average true regret and several percentile bounds. As above, we see that a smaller ϕ requires a smaller k to guarantee low MMR. It also illustrates the desirable anytime property of MMR: regret drops significantly with the "first few candidates" and levels off before reaching zero. For example, with $m = 10, n = 100, \phi = 0.6$, top-3 queries reduce MMR to 0.8 per voter from the MMR of 9 obtained with no queries; but an additional 3 candidates (i.e., top-6 queries) are needed to reduce regret from 0.8 per voter to 0. If we fix $\phi = 0.6$ and increase the number of candidates m, the k required for small MMR decreases in relation to m: we see that for $m = 5, 10, 20$ we need top-k queries with $k = 3, 6, 8$, respectively, to reach MMR of zero. This is, of course, specific to the Mallows model.

Figure 6.7 show histograms on two real-world data sets: *Sushi* [70] (10 alternatives and 5000 rankings) and *Dublin*, voting data from the Dublin North constituency in 2002 (12 candidates and 3662 rankings).[2] With *Sushi*, we divided the 5000 rankings into 50 voting profile instances, each with $n = 100$ rankings, and plotted MMR histograms using the same protocol as in Figure 6.1 and Figure 6.3; similarly, *Dublin* was divided into 73 profiles each with $n = 50$. Sushi results suggest that with top-5 queries one can usually find a necessary winner; but top-4 queries are usually enough to obtain low MMR sufficient for such a low-stakes group decision (i.e., what sushi to order). True regret histograms show the minimax solution is almost always the true winner. With *Dublin*, top-5 queries virtually guarantee MMR of no more than 2 per voter; top-6, MMR of 1 per voter; and top-7, MMR of 0.5 per voter. True regret plots show the minimax winner is either optimal or close to optimal in most profile instances.

[2]There are 43,942 ballots; 3662 are complete. See www.dublincountyreturningofficer.com

6.5 Conclusion

We have outlined a general framework for the design of multi-round elicitation protocols that are sensitive to tradeoffs between the number of rounds of elicitation imposed on voters, the amount of information elicited per round, and the quality of the proposed winner. Our framework is probabilistic, allowing one to account for realistic distributions of voter preferences and profiles. We have formulated a probabilistic method for choosing the ideal threshold k for top-k elicitation in one-round protocols, and developed an empirical methodology that applies to *any voting rule* and *any preference distribution*. While the method can be used purely heuristically, our PAC-analysis provides our methodology with statistical guarantees. Experiments on random Mallows models, as well as real-world data sets (sushi preferences and Irish electoral data) demonstrate the practical viability and advantages of our empirical approach.

There are numerous opportunities for future research. We have dealt mainly with one-round elicitation of top-k candidates—developing algorithms for optimal multi-round instantiations of our framework is an important next step. Critically, we must deal with posterior distributions that are generally intractable, though GRIM-based techniques as developed in Chapter 5 may help. We are also interested in more flexible query classes such as batched pairwise comparisons. While the empirical framework is applicable to any preference distribution, we still wish to analyze the performance on additional distributions, including more flexible mixture models. On the theoretical side, we expect that the PAC-analysis can be extended to different query classes and to multi-round protocols: probabilistic bounds on the amount of information required (e.g., k^* for top-k queries) should be significantly better than deterministic worst-case bounds [34] assuming, for example, a Mallows model. Bayesian approaches that assess candidate quality using expected regret rather than minimax regret are also of interest, especially in lower-stakes settings. We expect that combining expected regret and minimax regret might yield interesting solutions as well.

Chapter 7

Summary and Conclusions

In social choice, decisions must be made for a collection of agents, each of whom may have different preferences over the set of possible decisions. Social choice problems as well as their methodologies have become increasing important in our modern world as individuals and organizations have become more comfortable using Internet-enabled services through their connected devices (PCs, smartphones, etc.). Not only do applications of social choice abound, such as in meeting scheduling, activity planning and political voting, its techniques and methodologies have also had impact on ranking web search results, aggregating opinions, and systems for personalized product recommendations. However, many of the existing approaches in social choice require that agents think through and report their full preferences, usually in the form of rankings. This requirement not only imposes a cognitive burden for the agents, but also increases the communication cost, and violates their privacy by extracting potentially unnecessary personal information. This is in part why, for example, many political elections use the Plurality voting rule to elect candidates.

In this thesis, we have developed a framework for making robust social choice decisions given incomplete preference information from multiple agents. Moreover, our robust decision support framework naturally allows for preference elicitation schemes that aim to increase decision robustness.

Our framework aims to address the above issues. It is very general, encompassing any social choice mechanism that has a quantifiable social welfare interpretation, which includes many commonly studied and widely used voting rules. The minimax regret approach we take to computational social choice has been proven effective in single-agent settings [14, 13, 19] and in mechanism design [67]. By applying such decision-theoretically sound concepts with theoretically and practically effective algorithms that implement such concepts, including robust decision support and preference elicitation,

we show that many social choice problems can be practically solved while reducing the informational requirements of agent preferences and the computation requirements of robust optimization.

In particular, the key idea behind our framework is the notion of *minimax regret*. Minimax regret measures the best worst-case difference in the quality (i.e. social welfare) of an alternative compared to the unknown, but optimal, alternative. It assumes that partial preferences are completed in a worst-case fashion. Thus, while minimax regret is worst-case, and provides robust quality guarantees, the actual true regret may be significantly smaller. In many of our experiments, our preference elicitation algorithms are able to quickly drive down the minimax regret. This results in significant practical reduction in the amount of preference information required, compared to eliciting complete rankings. We also develop algorithms for learning heterogeneous models of preference rankings, which we then leverage in the context of *multi-round elicitation* schemes to further reduce the information elicited.

7.1 Chapter Summary of Main Results

Robust Optimization and Elicitation for Single-Choice Problems. In Chapter 3, we focus on how to make group decisions with only partial preference information. We introduce the concept of robust social choice, as defined by the notion of minimax regret, which provides the best worst-case guarantee with respect to the quality of the true, but unknown, winning alternative. We contrast minimax regret as a decision-criterion with possible and necessary winners, a previous solution concept, and show that it provides the more sensible foundation on which to implement winner determination with partial preferences.

We develop efficient, polynomial time algorithms for computing the minimax regret optimal alternative for a variety of common voting rules and show hardness results for certain other voting rules. We also developed incremental preference elicitation algorithms that identify an agent and pose a pairwise comparison query, in order to quickly reduce minimax regret. Experiments on real voting and preference data show that only a fraction of the full preference information is needed to identify an alternative with zero minimax regret (i.e., the true winning alternative). If small max regret values are tolerable, then even less information needs be elicited to find such a winner.

Robust Optimization and Elicitation for Multiple-Choice Problems. In Chapter 4, we focus on how to select a set, or *slate*, of alternatives given only partial preference

information. In particular, we study the popular scheme known as Proportional Representation (PR). PR is a natural selection criterion in which an agent *scores* a slate by the satisfaction that agent derives from the most preferred alternative within the slate as given by the agent's ranking. While it has recently been shown that PR is NP-hard even for the full information setting, greedy algorithms as well as integer programming formulations have been developed.

We again apply the minimax regret framework to selecting slates with only partial preference information. While the regret-based definitions naturally extend from single-winner to the multi-winner setting, computational results do not. However, we develop a greedy approach that is based on finding the next alternative that, when added to the current slate of alternatives, minimizes the maximum regret. We prove approximation guarantees for this greedy algorithm, including the fact that it reduces to the full information greedy algorithm when the marginal regret of adding an alternative to the slate is zero (for example, as is the case when full rankings are known). The preference elicitation algorithm that we devise exploits the conditional max regret from the greedy algorithm to drive the generation of a query, which consists of an agent and a pairwise comparison. Experiments on real datasets show the effectiveness of the greedy algorithm as well as that of the elicitation strategy, which quickly reduces regret of the minimax optimal slate.

Learning Rankings with Pairwise Preferences. In Chapter 5, we turn our focus to the problem of learning preference (ranking) distributions over alternatives. Not only is this problem important in recommender systems (for example, inferring missing preferences when given partial preference information), it allows us to tackle a variety of social choice problems, and in particular, design better preference elicitation strategies (as we explore in Chapter 6). While learning ranking distributions has been the subject of much research, no previous work has addressed the problem of learning distributions from pairwise comparisons, a basic building block of many preference structures. This chapter develops tractable learning algorithms for such pairwise comparison evidence. While we primarily develop algorithms for learning Mallows-based models (and mixtures thereof), we believe the approach can be extended to other distributions.

We introduce a key technical tool, a sampling methodology known as the *Generalized Repeated Insertion Model*, that can approximately sample Mallows models that have been conditioned on pairwise comparison evidence. While we do show that sampling and inference with conditional Mallows models is computationally difficult, our GRIM based sampler is exact for many specialized forms of preferences, including previously studied

forms (such as top- and bottom-k, partitioned, etc.). We also prove bounds on the approximation quality of GRIM-based samplers. We then use these samplers to devise EM-based learning algorithms to learn Mallows mixture models, calculate model log-likelihoods, and perform non-parametric estimation. Experiments on real-world datasets demonstrate the scalability and effectiveness of the algorithms, as well as interesting clustering patterns in the Movielens and sushi datasets.

Elicitation with Probabilistic Preference Distributions. Finally, in Chapter 6, we circle back to the preference elicitation problem in social choice, but from a probabilistic, and multi-round, perspective. In this chapter, we discuss the issues inherent in fully incremental schemes. In practice, they may cause substantial, and unnecessary, delay until a decision is made—they require agents to respond before asking the next query, and hence any blocking agent significantly delays the outcome. Furthermore, they create interruption costs which occur when an agent must wait for other agents to respond to their respective queries before that agent is queried again. Such issues are practically important when implementing a group decision problem, especially in a time-sensitive setting.

In this chapter, we propose a probabilistic, multi-round model of vote elicitation that combines advantages of incremental elicitation (i.e., minimizing the total amount of information elicited) with those of batch or full information schemes (where we send non-blocking, or non-dependent queries to selected agents in a few *rounds*). The multi-round model specifically accounts for rounds, where in each round the elicitor sends non-blocking queries to a set of agents. We exploit either previously known or learned distributions of preferences (for example, obtained from algorithms in Chapter 5) to infer the ideal number of rounds and the ideal queries within a round, so that a winning alternative (or an alternative with small minimax regret) exists with high probability. In particular, we analyze a one-round top-k query protocol where we demonstrate how the best k is chosen. We provide a PAC-style sample complexity analysis which provides confidence guarantees on the suggested k.

Our empirical results not only show the effectiveness of our one-round protocol, but also provide a practical procedure under which one can select the appropriate elicitation parameters (i.e. the value of k).

7.2 Contributions

We summarize our thesis contributions in point form below.

Chapter 3

- Defining the *minimax regret (MMR)* criterion for making robust social choice decisions with partial preference information.
- Comparing MMR to possible and necessary winners, showing it generalizes on necessary winners and showing advantages including worst-case guarantees, and the possibility of alternatives that are not possible winners but have good max regret.
- Offering basic observations about the complexity of computing MMR optimal alternatives.
- Developing polynomial time algorithms for computing MMR for common voting rules including positional scoring, maxmin fairness, maximin and Bucklin.
- Developing an elicitation strategy known as current solution strategy (CSS) for pairwise and top-k queries that quickly reduces MMR.
- Experiments on real and synthetic datasets that show CSS is superior to a "volumetric" strategy and random querying for both pairwise and top-k queries. Experiments also show that, in practice, only a fraction of preference information is required to make optimal or near-optimal decisions.

Chapter 4

- Defining MMR for Proportional Representation, a slate optimization problem.
- Defining the *additional alternative problem*, which is a key component of a greedy algorithm for approximating the MMR optimal slate.
- Empirical results showing the partial information greedy algorithm produces optimal or near optimal slates in practice.
- Developing a preference elicitation strategy that quickly reduces MMR of the optimal slate (or the max regret of slate generated by our greedy approximation algorithm) with a desirable anytime profile.

Chapter 5

- Development of a new class of rank sampling algorithms called *generalized repeated insertion model* (GRIM).

- Development of a GRIM-based conditional Mallows model sampler.
- Proving $\#P$-hardness of conditional Mallows inference.
- Proving upper and lower bounds on the approximation quality of our conditional Mallows sampler.
- Proving that our sampler is exact for a widely studied class of preference structures.
- Development of an ergodic MCMC sampler for conditional Mallows.
- Development of algorithms for estimating log-likelihood of a Mallows mixture model.
- Development of Monte Carlo EM algorithm for learning a mixture of Mallows models given only agents' pairwise comparisons (the first such algorithm).
- Experiments on synthetic, Movielens, and sushi data reveal our proposed learning algorithm is reasonably efficient, that it indeed learns interesting clusters of rankings in the datasets, and that it has reasonable predictive capabilities.

Chapter 6

- Introduction of a framework for trading off the number of rounds of elicitation with the total information revealed to obtain a (near) optimal MMR solution.
- Using probabilistic preference models to analyze the best elicitation strategy for one-round top-k queries.
- Development of a general Monte Carlo methodology to find the smallest value of k needed in one-round top-k elicitation for a given MMR tolerance, and theoretical PAC-bounds on sample complexity.
- Providing empirical results on real data that show effectiveness of the methodology and the relatively small value of k needed to realize near-optimal winners.

7.3 Future Work

Multi-attribute Elicitation. While our results suggest that incremental elicitation is viable in many practical domains, a number of interesting avenues for future research remain. Apart from developing computational and elicitation schemes for additional single- and multi-winner voting rules, one important direction is to develop approaches

well-suited for multi-attribute domains, where alternatives and/or voters are specified using particular instantiations of attributes, and preferences are represented as compact functions of these attributes. For example, alternatives may be cars and each car can be represented by a vector of attributes such as color, brand, year, etc. An agent's preference or utility function over a car may for example, be a linear combination of relevant attributes. A social choice problem in this context, may be for a car manufacturer to find a car design (an instantiation of attribute values) that will maximally satisfy all agents. One can also use a more flexible class called *Generalized Additive Utility* [17] to model more intricate utility functions. These are also known as *combinatorial domains*, where the number of alternatives can be exponentially large in the number of attributes. For an overview of voting over combinatorial domains, see Brandt et al. [16, Chapter 9]. Elicitation strategies and robust optimization methods must account for a much bigger outcome space and exploit the utility and attribute representation that can help in computing MMR and in generating helpful queries. Such models are particularly relevant in recommender systems and product configuration.

Query Types. While we have worked primarily with pairwise comparisons and top-k queries, there are a few other query types that can be natural for human agents to answer. This includes selecting the most preferred alternative from a subset of alternatives. Answering such a query would generate $k-1$ pairwise comparisons if k is the size of the subset. One can also request the agent to rank the alternatives of the subset, thereby revealing even more preference information. One can also ask for the top-t items, the bottom-t items and so on. Elicitation may also be natural with questions regarding both most preferred and least preferred items. For example, a user may clearly identify the most preferred restaurant and the least preferred restaurant (e.g., a place that is far from home). Such information can be quite informative in eliminating and narrowing the set of candidate winners. Perhaps not as natural, one can also ask "rank position" queries that cannot be represented with pairwise comparisons. These include "what is an upper/lower bound on the rank position of alternative a?" Or "what is the rank position of a?" These would require different algorithms for computing minimax regret, unlike ones we developed for pairwise preferences. It might also be interesting to mix these queries.

Moving away from ranking-based preferences to real-valued ones (i.e., corresponding to utility values), these typically allow voters to express numerical values for each alternative. Special examples include approval, cumulative and range voting. Recall that Kalech et al. [69] provide a simple elicitation algorithm for range voting, where queries are of the form "what is your next highest preferred candidate, and what is its utility?"

One can imagine more nuanced queries such as "what is the utility of alternative a?", or "give upper and/or lower bounds on the utility of a," or even pairwise comparisons. There might also be constraints, as in cumulative voting where the sum of all utilities must equal one, which can create consistency issues as well as implications of the unknown utilities (e.g., if every utility is known except for one alternative, then utility sum constraint implies its value).

Robust Optimization and Elicitation for Rank Aggregation. While we have presented algorithms for both single- and multi-winner schemes, we have not addressed rank aggregation schemes. Although it appears to be less prevalent, studying rank aggregation as a decision problem with partial preferences is an interesting research direction. This makes sense if the consensus ranking, and not parts of it, has a useful purpose. An example where a consensus ranking is useful is in graduate program admissions where an aggregate ranking needs to be produced from a committee of evaluators. In practice, not all evaluators can read all applications and thus a consensus ranking must be produced from only partial rankings from the evaluators. The Duke University computer science program uses a variation of the Kemeny ranking to solve this exact problem [31]. As another example, Lu and Boutilier [82] developed the unavailable candidate model, which justifies a consensus ranking as a decision policy, that selects as a winner the highest ranking candidate that is available *a posteriori*. Under this model, the Kemeny ranking can be justified as the solution to a special case of the model when the probability of candidates being unavailable approaches 1. It's likely that in the deterministic worst-case, Kemeny would require all voters to give entire rankings. Nevertheless, weighted versions of Kemeny, where emphasis is placed on getting top of the ranking correct, may require less information when one allows a small minimax regret tolerance.

Robust and Probabilistic Optimization. While the minimax regret criterion offers a robust approach to group decisions with quantifiable guarantees, it also has drawbacks. First, the guarantees are worst-case and provide no information about what the true regret might be. In our experiments on Irish and Sushi datasets, we see that the true regret is typically much smaller than the minimax regret. The fact is that real-world preferences are rarely worst-case instances for the elicitation problem. We can also exploit known probabilistic information regarding populational preferences to make even better decisions. Knowing partial preference information, one can certainly condition a learned populational model (e.g., a Mallows mixture model) and infer the most probable outcome (MAP estimate). But more interesting analyses would mix probabilistic and regret-based

reasoning. For example, of all alternatives that have relatively small max regret, one might recommend the winning alternative that is most probable under the posterior distribution.

Another approach is to use *expected regret*. Assume a distribution P over preference profiles and some observation model L (i.e., a distribution which conditioned on a preference profile generates an partial profile). Then the posterior is of the form $P(\mathbf{v}|\mathbf{p}) \propto L(\mathbf{p}|\mathbf{v})P(\mathbf{v})$, where $\mathbf{v}$ and $\mathbf{p}$ is a complete and partial preference profile, respectively. Now given $\mathbf{p}$ and a voting rule r, a Bayesian decision-maker would be interested in finding the alternative (or a slate) with smallest *expected regret*

$$\max_{a \in A} \mathop{\mathbb{E}}_{P(\mathbf{v}|\mathbf{p})} [s(r(a), \mathbf{v}) - s(a, \mathbf{v})] \; . \tag{7.1}$$

Alternatively, one may also be interested in maximizing expected score, $\max_{a \in A} \mathbb{E}_{P(\mathbf{v}|\mathbf{p})} s(a, \mathbf{v})$. The benefits of the Bayesian approach include better estimates of the actual regret, incorporating prior knowledge that weights more realistic profiles higher than extreme worst-cases, and being able to average regret over all (consistent and complete) profiles. The Bayesian approach would be more applicable in low-stake settings, since there is no guarantee that the prior P accurately reflects the underlying preference distribution (although P can be learned from historical data), and therefore no strong guarantees can be provided that guards against a worst-case event, in contrast to minimax regret.

Extending Pairwise Preference Learning Algorithms. One of the weaknesses of the Mallows model is its lack of flexibility in various dimensions, such as allowing different dispersion "rates" in different regions of the ranking. Models that allow more flexibility while controlling for overfitting could lead to more realistic ranking models for real-world settings. Other extensions include exploration of other probabilistic models of incomplete preferences that employ different distributions over rankings, such as Plackett-Luce or weighted Mallows; that account for noisy comparison data from users; and that account for data that is not missing at random—this may occur, say, in settings in which a bias exists towards observing preferences for higher ranked alternatives. The ideas behind the sampling algorithms for conditional preference distributions can also be extended to other models such as Plackett-Luce, the Thurstonian model, and even the recent riffled independence [61] model, using the same constrained sampling approach. Given an appropriate observation model, it can also be extended to noisy observations, when pairwise comparisons are intransitive.

Modelling Single-Peaked Preference Distributions. One interesting direction is the modelling of single-peaked preference profiles. These are profiles in which there exists an ordering O of A (picture this as placing items in a line from left to right) such that any user's preference has a peak point a^* (i.e., most preferred item) and for x, y on the left (right) side of a^*, if x is left (right) of y in O, then $y \succ x$. There is no obvious probabilistic model that generates single-peaked profiles. But recent work on riffled independence provides an interesting approach to this problem. Suppose we have some ordering O of A, we generate a single-peaked preference ranking by first choosing the peak, or the most preferred item, at random. Then, the remaining items can be partitioned into the left and right most items. The ranking of the left (and right) most items are determined according to how far they are from the peak. However, that does not produce a full ranking r, which must have the peak at the top position and the left and right most items must be interleaved in such a way that r restricted to the left (right) most items is consistent with O. For example suppose $O = abcde$ and c is the peak. Then if the left items are ranked ba and the right items are ranked de, then $cdbae$ is a valid ranking but not $cdabe$ since $b \succ a$. The interleaving process is performed randomly, for example, as in a riffle shuffle of two decks of cards.

Multi-Round, Probabilistically Guided Elicitation. We have dealt mainly with one-round elicitation of top-k candidates—developing algorithms for optimal multi-round instantiations of our framework is an important next step. Critically, we must deal with posterior distributions that are generally intractable, though GRIM-based techniques may help. It would also be interesting to explore more flexible query classes such as batched pairwise comparisons. While the empirical framework is applicable to any preference distribution, we still wish to analyze the performance on additional distributions, including more flexible mixture models. On the theoretical side, we expect our PAC-analysis can be extended to different query classes and to multi-round protocols: we expect that probabilistic bounds on the amount of information required (e.g., k^* for top-k queries) will be significantly better than deterministic worst-case bounds [34] assuming, for example, a Mallows model.

Bibliography

[1] Shivani Agarwal and Dan Roth. Learnability of bipartite ranking functions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory (COLT-05)*, pages 16–31, Bertinoro, Italy, 2005. 92

[2] Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 415–424. SIAM, 2007. 25

[3] Hassene Aissi, Cristina Bazgan, and Daniel Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197(2):427–438, 2009. 56

[4] Kenneth J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58:328–346, August 1950. 3, 14, 22, 23

[5] Igor Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, 27:57–65, 2000. 56

[6] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. Justified representation in approval-based committee voting. In *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, Austin, TX, 2015. 21

[7] Yoram Bachrach, Nadja Betzler, and Piotr Faliszewski. Probabilistic possible winner determination. In *Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 697–702, Atlanta, GA, 2010. 28

[8] John Bartholdi III, Craig Tovey, and Michael Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2): 157–165, 1989. 24, 27, 114

[9] Christophe Biernacki and Julien Jacques. A generative model for rank data based on insertion sort algorithm. *Computational Statistics and Data Analysis*, 58:162–176, 2013. 93

[10] Duncan Black. On the rationale of group decision-making. *Journal of Political Economy*, 56(1):23–34, 1948. 11

[11] Craig Boutilier. Computational decision support: Regret-based models for optimization and preference elicitation. In P. H. Crowley and T. R. Zentall, editors, *Comparative Decision Making: Analysis and Support Across Disciplines and Applications*, pages 423–453. Oxford University Press, Oxford, 2013. 56

[12] Craig Boutilier, Fahiem Bacchus, and Ronen I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 56–64, Seattle, 2001. 5, 56

[13] Craig Boutilier, Tuomas Sandholm, and Rob Shields. Eliciting bid taker non-price preferences in (combinatorial) auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 204–211, San Jose, CA, 2004. 5, 34, 36, 56, 150

[14] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artifical Intelligence*, 170(8–9):686–713, 2006. 5, 34, 36, 46, 56, 150

[15] Craig Boutilier, Kevin Regan, and Paolo Viappiani. Online feature elicitation in interactive optimization. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML-09)*, pages 73–80, Montreal, 2009. 5, 56

[16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, Cambridge, 2015. To appear. 10, 13, 156

[17] Darius Braziunas and Craig Boutilier. Preference elicitation and generalized additive utility (nectar paper). In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, Boston, 2006. 156

[18] Darius Braziunas and Craig Boutilier. Minimax regret-based elicitation of generalized additive utilities. In *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 25–32, Vancouver, 2007. 5, 56

[19] Darius Braziunas and Craig Boutilier. Assessing regret-based preference elicitation with the UTPREF recommendation system. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC'10)*, pages 219–228, Cambridge, MA, 2010. 5, 34, 36, 46, 56, 150

[20] Graham Brightwell and Peter Winkler. Counting linear extensions is #P-complete. In *ACM Symposium on the Theory of Computing*, pages 175–181, New Orleans, LA, 1991. 28, 97, 110

[21] Chris J.C. Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, July 2010. 78, 92

[22] Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. Learning to rank using gradient descent. In *Proceedings of the Twenty-second International Conference on Machine Learning (ICML-05)*, pages 89–96, Bonn, Germany, 2005. 92

[23] Róbert Busa-Fekete, Eyke Hüllermeier, and Balázs Szörényi. Preference-based rank elicitation using statistical models: The case of Mallows. In *Proceedings of the Thirty-first International Conference on Machine Learning (ICML-14)*, pages 1071–1079, Beijing, China, 2014. 92

[24] Ludwig M. Busse, Peter Orbanz, and Joachim M. Buhmann. Cluster analysis of heterogeneous rank data. In *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML-07)*, pages 113–120, Corvallis, OR, 2007. 6, 7, 81, 82, 93, 114

[25] John R. Chamberlin and Paul N. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *The American Political Science Review*, 77(3):718–733, 1983. 6, 20, 59

[26] Weiwei Cheng, Krzysztof Dembczynski, and Eyke Hüllermeier. Label ranking methods based on the Plackett-Luce model. In *Proceedings of the Twenty-seventh International Conference on Machine Learning (ICML-10)*, pages 215–222, Haifa, Israel, 2010. 12, 79, 81, 82

[27] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-07)*, pages 51–69, Harrachov, Czech Republic, 2007. 10, 13

[28] Reuven Cohen and Liran Katzir. The generalized maximum coverage problem. *Inf. Process. Lett.*, 108(1):15–22, 2008. 21

[29] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999. 92

[30] marquis de (Marie Jean Antoine Nicolas de Caritat) Condorcet. *Essai sur l'Application de l'Analyse a la Probabilite des Decisions rendues a la Probabilite des Voix*. Paris: L'Imprimerie Royale, 1785. 15, 83

[31] Vincent Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010. 10, 157

[32] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pages 392–397, Edmonton, 2002. 4, 26, 27, 28, 46, 55

[33] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 145–152, Edinburgh, 2005. 92

[34] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the Sixth ACM Conference on Electronic Commerce (EC'05)*, pages 78–87, Vancouver, 2005. 29, 55, 149, 159

[35] Vincent Conitzer, Andrew J. Davenport, and Jayant Kalagnanam. Improved bounds for computing kemeny rankings. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 620–626, Boston, 2006. 24

[36] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007. 29

[37] Vincent Conitzer, Matthew Rognlie, and Lirong Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 109–115, Pasadena, CA, 2009. 92

[38] Don Coppersmith, Lisa Fleischer, and Atri Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the*

Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-06), pages 776–782, Miami, FL, 2006. 24, 117

[39] Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781. 3, 16

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. 111

[41] Persi Diaconis. *Group Representations in Probability and Statistics*, volume 11 of *Lecture Notes–Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1988. 12, 117

[42] Ning Ding and Fangzhen Lin. Voting with partial information: What questions to ask? In *Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-13)*, pages 1237–1238, St. Paul, MN, 2013. 56

[43] Jean-Paul Doignon, Aleksandar Pekec, and Michel Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69(1):33–54, 2004. 79, 80, 86, 87, 88

[44] Joanna Drummond and Craig Boutilier. Elicitation and approximately stable matching with partial preferences. In *Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence (IJCAI-13)*, pages 97–105, Beijing, 2013. 46

[45] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International World Wide Web Conference (WWW-01)*, pages 613–622, Hong Kong, 2001. ACM. 23, 24, 114

[46] Yuval Filmus and Joel Oren. Efficient voting via the top-k elicitation scheme: A probabilistic approach. In *Proceedings of the Fifteenth ACM Conference on Electronic Commerce (EC'14)*, pages 295–312, Palo Alto, 2014. 134, 140

[47] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):359–369, 1986. 82

[48] Michael A. Fligner and Joseph S. Verducci, editors. *Probability Models and Statistical Analysis for Ranking Data*, volume 80 of *Lecture Notes in Statistics*. New York: Springer-Verlag, 1993. 82

[49] Brian Francis, Regina Dittrich, Reinhold Hatzinger, and Les Humphreys. A mixture model for longitudinal partially ranked data. *Communications in Statistics — Theory and Methods*, 43(4):722–734, 2014. 93

[50] Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003. 92

[51] Wulf Gaertner. Domain restrictions. In Kenneth J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 131–170. North Holland, 2002. 3

[52] Wulf Gaertner. *A Primer in Social Choice Theory*. LSE Perspectives in Economic Analysis. Oxford University Press, August 2006. 10

[53] Zoubin Ghahramani and Mike I. Jordan. Learning from incomplete data. Technical report, Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab, 1995. 90

[54] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973. 14, 15

[55] Isobel Claire Gormley and Thomas Brendan Murphy. A latent space model for rank data. In Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, Anna Goldenberg, Eric P. Xing, and Alice X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, pages 90–102. Springer, 2007. 82

[56] Isobel Claire Gormley and Thomas Brendan Murphy. Exploring voting blocs within the Irish electorate. *Journal of the American Statistical Association*, 103(483): 1014–1027, 2008. 93

[57] Pranava R. Goundan and Andreas S. Schulz. Revisiting the greedy approach to submodular set function maximization. *Optimization Online*, 2007. URL `http://www.optimization-online.org/DB_HTML/2007/08/1740.html`. 68, 69

[58] John Guiver and Edward Snelson. Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML-09)*, pages 377–384, Montreal, 2009. 12, 79, 81, 82, 93

[59] Mohammad Taghi Hajiaghayi, Mohammad Mahdian, and Vahab S. Mirrokni. The facility location problem with general cost functions. *Networks*, 42(1):42–47, 2003. 21

[60] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In Peter J. Bartlett, Bernhard Schölkopf, Dale Schuurmans, and Alex J. Smola, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000. 92

[61] Jonathan Huang and Carlos Guestrin. Riffled independence for ranked data. In *Advances in Neural Information Processing Systems 21*, pages 799–807, Vancouver, 2009. 93, 158

[62] Jonathan Huang, Carlos Guestrin, and Leonidas Guibas. Fourier theoretic probabilistic inference over permutations. *Journal of Machine Learning Research*, 10: 997–1070, May 2009. 7

[63] Jonathan Huang, Ashish Kapoor, and Carlos Guestrin. Riffled independence for efficient inference with partial ranking. *Journal of Artificial Intelligence Research*, 44:491–532, 2012. 94

[64] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16–17):1897–1916, 2008. 92

[65] Nathanaël Hyafil and Craig Boutilier. Regret-based incremental partial revelation mechanisms. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 672–678, Boston, 2006. 46, 56

[66] Nathanaël Hyafil and Craig Boutilier. Regret-based incremental partial revelation mechanisms. In *Proceedings of the Twenty-second National Conference on Artificial Intelligence (AAAI-07)*, pages 72–78, Vancouver, 2006. 46, 56

[67] Nathanaël Hyafil and Craig Boutilier. Mechanism design with partial revelation. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1333–1340, Hyderabad, India, 2007. 34, 46, 56, 150

[68] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 133–142, Edmonton, AB, 2002. 92

[69] Meir Kalech, Sarit Kraus, Gal A. Kaminka, and Claudia V. Goldman. Practical voting rules with partial information. *Journal of Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, 2011. 29, 31, 50, 51, 55, 135, 156

[70] Toshihiro Kamishima, Hideto Kazawa, and Shotaro Akaho. Supervised ordering: An empirical survey. In *IEEE International Conference on Data Mining (ICDM-05)*, pages 673–676, Houston, TX, 2005. 51, 121, 148

[71] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959. 24, 84

[72] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 12

[73] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *ACM STOC*, pages 95–103, New York, NY, 2007. ACM. 24

[74] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. *Journal of the ACM*, 51:263–280, 2004. 22, 59

[75] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, Edinburgh, 2005. 26, 27, 28

[76] Kathrin Konczak and Jerome Lang. Voting procedures with incomplete preferences. In *IJCAI Workshop on Advances in Preference Handling*, pages 124–129, 2005. 4

[77] Panos Kouvelis and Gang Yu. *Robust Discrete Optimization and Its Applications*. Kluwer, Dordrecht, 1997. 56

[78] Jérôme Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1366–1371, Hyderabad, India, 2007. 4

[79] Guy Lebanon and Yi Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008. 7, 12, 78, 80, 82, 93, 98, 101, 127, 129

[80] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. 116

[81] Jordan J. Louviere, David A. Hensher, and Joffre D. Swait. *Stated Choice Methods: Analysis and Application*. Cambridge University Press, Cambridge, 2000. 79, 81

[82] Tyler Lu and Craig Boutilier. The unavailable candidate model: A decision-theoretic view of social choice. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC'10)*, pages 263–274, Cambridge, MA, 2010. 23, 157

[83] Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 280–286, Barcelona, 2011. 6, 19, 21, 59, 64, 67, 69

[84] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 287–293, Barcelona, 2011. 63, 137

[85] R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959. 78, 82

[86] Brendan Lucier and Joel Oren. Online (budgeted) social choice. In *Proceedings of the Fourth International Workshop on Computational Social Choice (COMSOC-2012)*, Kraków, Poland, 2012. 21

[87] Colin L. Mallows. Non-null ranking models. In *Biometrika*, volume 44, pages 114–130, 1957. 7, 78, 82, 83, 84, 134

[88] John I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall, London, 1995. 6, 78, 82, 83, 134

[89] Benjamin M. Marlin and Richard S. Zemel. Collaborative filtering and the missing at random assumption. In *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 50–54, Vancouver, 2007. 90

[90] Marina Meila and Harr Chen. Dirichlet process mixtures of generalized Mallows models. In *Proceedings of the Twenty-sixth Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 358–367, Catalina Island, CA, 2010. 6, 7, 82, 93

[91] Reshef Meir, Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research (JAIR)*, 33:149–178, 2008. 19

[92] Burt L. Monroe. Fully proportional representation. *The American Political Science Review*, 89(4):925–940, 1995. 6, 21, 59

[93] Hervé Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35(4): 437–455, 1980. 11

[94] Eitan Muller and Mark A. Satterthwaite. The equivalence of strong positive association and strategy-proofness. *Journal of Economic Theory*, 14:412–418, 1977. 13, 14

[95] Thomas Brendan Murphy and Donal Martin. Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis*, 41:645–655, January 2003. 6, 7, 90, 93

[96] Radford Neal and Geoff Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, Cambridge, MA, 1999. 116

[97] Hannu Nurmi. *Voting Procedures Under Uncertainty*. Springer-Verlag, Berlin, 2002. 10

[98] Christos H. Papadimitriou, Michael Schapira, and Yaron Singer. On the hardness of being truthful. In *49th Annual IEEE Symposium on Foundations of Computer Science, (FOCS-08)*, pages 250–259, Philadelphia, 2008. 21

[99] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Incompleteness and incomparability in preference aggregation. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1464–1469, Hyderabad, India, 2007. 27, 28

[100] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Aggregating partially ordered preferences. *Journal of Logic and Computation*, 19: 475–502, 2009. 34

[101] R. L. Plackett. The analysis of permutations. *Applied Statistics*, 24:193–202, 1975. 78, 82

[102] Richard F. Potthoff and Steven J. Brams. Proportional representation: Broadening the options. *Journal of Theoretical Politics*, 10(2):147–178, 1998. 20

[103] Ariel D. Procaccia, Jeffrey S. Rosenschein, and Aviv Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30:353–362, 2008. 21, 63

[104] Kevin Regan and Craig Boutilier. Regret-based reward elicitation for Markov decision processes. In *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 454–451, Montreal, 2009. 56

[105] Kevin Regan and Craig Boutilier. Eliciting additive reward functions for markov decision processes. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 2159–2164, Barcelona, 2011. 56

[106] Michel Regenwetter, Bernard Grofman, A. A. J. Marley, and Ilia Tsetlin. *Behavioral Social Choice: Probabilistic Models, Statistical Inference, and Applications*. Cambridge University Press, Cambridge, 2006. 54

[107] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. A short introduction to preferences: Between artificial intelligence and social choice. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(4):1–102, 2011. 13

[108] Cynthia Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, 2009. 92

[109] Ahti Salo and Raimo P. Hämäläinen. Preference ratios in multiattribute evaluation (PRIME)—elicitation and decision procedures under incomplete information. *IEEE Trans. on Systems, Man and Cybernetics*, 31(6):533–545, 2001. 5, 56

[110] Mark A. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975. 14, 15

[111] Grace S. Shieh. A weighted Kendall's tau statistic. *Statistics and Probability Letters*, 39(1):17–24, 1998. 89

[112] Piotr Skowron, Piotr Faliszewski, and Arkadii Slinko. Achieving proportional representation is easy in practice. In *Proceedings of the Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-13)*, pages 399–406, St. Paul, MN, 2013. 21

[113] Warren Smith. Range voting. `http://www.math.temple.edu/~wds/homepage/rangevote.pdf`, 2000. 18, 34

[114] Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Preference elicitation for general random utility models. In *Proceedings of the Twenty-ninth Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 596–605, Bellevue, WA, 2013. 56

[115] Olivier Toubia, John R. Hauser, and Duncan I. Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1): 116–131, 2004. 50

[116] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984. 139

[117] Merijn van Erp and Lambert Schomaker. Variants of the borda count method for combining ranked classifier hypotheses. In *Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, pages 443–452, Amsterdam, NL, 2000. 23

[118] Paolo Viappiani and Craig Boutilier. Regret-based optimal recommendation sets in conversational recommender systems. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys09)*, pages 101–108, New York, 2009. 46, 56

[119] Maksims Volkovs and Richard S. Zemel. Boltzrank: Learning to maximize expected ranking gain. In *Proceedings of the Twenty-sixth International Conference on Machine Learning (ICML-09)*, page 137, Montreal, 2009. 92

[120] Maksims Volkovs and Richard S. Zemel. Efficient sampling for bipartite matching problems. In *Advances in Neural Information Processing Systems 25 (NIPS-12)*, pages 1322–1330, Lake Tahoe, NV, 2012. 93

[121] Toby Walsh. Complexity of terminating preference elicitation. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-08)*, pages 967–974, Estoril, Portugal, 2008. 4, 29, 46, 55

[122] Tianhan Wang and Craig Boutilier. Incremental utility elicitation with the minimax regret decision criterion. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 309–316, Acapulco, 2003. 5, 56

[123] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the Twenty-third*

AAAI Conference on Artificial Intelligence (AAAI-08), pages 202–207, Chicago, 2008. 4, 27

[124] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011. 35, 36

[125] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on the Theory of Computing*, pages 209–213, Atlanta, 1979. 29

[126] Peyton Young. Optimal voting rules. *Journal of Economic Perspectives*, 9:51–64, 1995. 78, 84, 92