Computational Voting Theory: Of the Agents, By the Agents, For the Agents

A thesis submitted for the degree of
"Doctor of Philosophy"

by

Ariel D. Procaccia

Submitted to the Senate of the Hebrew University

September 2008

This work was carried out under the supervision of

Professor Jeffrey S. Rosenschein

Acknowledgments

First, I would like to thank my advisor, Jeff Rosenschein, for his unwavering support. He is honestly one of the kindest persons I know. Jeff, thank you for providing me with everything I needed in order to develop myself as a researcher and a teacher. Equally importantly, thank you for many incredibly fun hours of chatting about musicals, movies, dogs, family, and whatnot.

I was fortunate to work with many excellent coauthors: Yoram Bachrach, Iannis Caragiannis, Vince Conitzer, Jason Covey, Shahar Dobzinski, Michal Feldman, Felix Fischer, Chris Homan, Christos Kaklamanis, Gal Kaminka, Nikos Karanikolas, Vangelos Markakis, Reshef Meir, Bezalel Peleg, Yoni Peleg, Jeff Rosenschein, Amin Saberi, Alex Samorodnitsky, Lirong Xia, Aviv Zohar, and Michael Zuckerman.

I also want to thank my friends at Hebrew U for making my time here so pleasant.

At this point it is common practice to write something like "I thank my wife, without whom I would not have survived my graduate studies". Alas, I have enjoyed my graduate studies. But Vera, thank you for making my life so much richer.

Abstract

The mathematical investigation of voting originated in the 17th century with such pioneers as the marquis de Condorcet and the chevalier de Borda, was taken up in the 18th century by Lewis Carroll, and exploded in the second half of the 20th century with the foundational work of Noble laureate Kenneth Arrow. Broadly speaking, the mathematical model of voting deals with a set of n agents that must reach a collective decision with respect to a set of m alternatives. Each agent submits a ranking of the alternatives; the outcome is then decided by a social choice function.

The field of Multi-Agent Systems is much younger, barely two decades old, and is concerned with systems occupied by multiple heterogeneous, autonomous, and self-interested agents. Collective decision making in such systems is one of the prominent and most challenging issues. Fortunately, the centuries of work on Voting Theory can be leveraged to reach a consensus among agents, but applying voting in distributed computational settings requires a richer understanding of the computational aspects of voting. Achieving such an understanding is the goal of this thesis.

We present our results on Computational Voting Theory in three parts, as follows.

Part I: Elections and Approximation. The first part focuses on using the paradigm of approximation, so common in the theory of computer science, to obtain novel positive results with respect to Voting. Chapter 3 deals with the social choice function suggested by Lewis Carroll. It has been known for some time that it is computationally intractable to determine the score of an alternative under this rule, and consequently hard to determine the winner of the election. Our main result in this context is a randomized rounding algorithm that yields an $\mathcal{O}(\log m)$ approximation ratio.

In Chapter 4, we apply the concept of approximation to a different classic problem. Voting trees describe an iterative procedure for selecting a single vertex from a tournament. It has long been known that there is no voting tree that always singles out a vertex with maximum degree. We study the power of voting trees in approximating the maximum degree. We give upper and lower bounds on the worst-case ratio between the degree of the vertex chosen by a tree and the maximum degree, both for the deterministic model concerned with a single fixed tree, and for randomizations over arbitrary sets of trees.

Part II: Elections and Computational Learning. The second part of the thesis studies the interplay between computational learning theory and voting theory. Chapter 5 investigates the learnability of two classes of social choice functions, as functions from the preferences of the agents to alternatives. We find that one of the two classes is efficiently learnable, whereas the other is harder to learn. We apply our results in an emerging theory: automated design of voting rules by learning.

Chapter 6 takes a step forward towards establishing a theory of incentives in a general machine learning framework. We focus on a game-theoretic regression learning setting where private information is elicited from multiple agents, which are interested in different distributions over the sample space; this conflict potentially gives rise to untruthfulness on the part of the agents. We show that various positive results can be obtained. Our techniques rely on classic concepts from social choice theory such as single peaked preferences, hence our results are intimately related to Voting Theory.

Part III: Frequency of Manipulation in Elections. The third and final part of thesis provides an analysis of the frequency of manipulation in elections. A well-known impossibility result asserts that any "reasonable" social choice function is prone to manipulation by the agents, that is, an agent can benefit by lying in certain situations. It has been proposed that computational hardness might prove a barrier against manipulation. In Chapter 7 we present analytic results that suggest that manipulation may be tractable under typical distributions on the preferences of the agents, even under social choice functions that are hard to manipulate in the worst-case.

In Chapter 8, we analyze the probability that a coalition of manipulators is able to sway the outcome of the election. Our theorems establish a threshold phenomenon: for many typical distributions, this probability is very small if the size of the coalition is below a certain threshold, and close to one if the size of the coalition is above the threshold.

Ultimately, we advocate certain agendas that all involve using computer science paradigms to obtain novel, positive results in some of the classic problems of Voting Theory.

Contents

Chapter 1

Introduction

Throughout the thesis we advocate several agendas, but hopefully one of the important contributions of this thesis will be coining the term "Computational Voting Theory", and positioning this field as a strict subset of the area called "Computational Social Choice". In the sequel we elaborate on this point. We note that this chapter overlaps with portions of the presentation given in subsequent chapters, but here we strive to provide the reader with a bird's eye view of the field. The exact definitions of many of the terms informally mentioned in this chapter can be found in Chapter 2.

1.1 A Broad Overview of Computational Voting Theory

Before we begin our discussion of Computational Voting Theory, it seems appropriate to briefly present the broader field of Computational Social Choice. In general, Social Choice Theory is concerned with the design and analysis of methods for collective decision making. This field has been, for several centuries now, the object of investigation by mathematicians and economists.

In the last two decades computer scientists, and especially researchers in Artificial Intelligence (AI), have become increasingly interested in Computational Social Choice. The attention is stimulated by the fact that Social Choice techniques have been shown to facilitate the design and analysis of Multi-Agent Systems (MAS). Indeed, MAS are often decentralized and populated by heterogeneous, self-interested agents—exactly the type of entities generally studied in economics! Curiously, Social Choice paradigms that often fail to capture human interactions are more applicable when the agents are rational software programs.

Computational Social Choice deals with, but is not necessarily limited to, the following areas [24]:

1. *Fair Division.* This area deals with the allocation of goods among self-interested agents, in a way that satisfies different desiderata. Examples of desiderata are: *Pareto efficiency*, meaning that no other allocation is weakly preferred by all agents and strongly preferred by some; and *envy-freeness*, in the sense that no agent prefers the bundle given to another agent. See the survey by Chevaleyre et al. [23] for more information.

2. *Coalition Formation.* In many settings agents cooperate in order to achieve common goals. Since the agents are assumed to be self-interested, game theorists have sought criteria for stable coalition structures, i.e., structures such that no agent has an incentive to deviate from

its assigned coalition. Computer scientists have examined the algorithmic aspects of coalition formation (see, e.g., Sandholm et al. [134]).

3. *Judgment Aggregation.* This field deals with the aggregation of agents' judgements on interconnected propositions into collective judgements, and is closely related to Voting Theory.

4. *Computational Voting Theory.* This is the subject of the remainder of this section, and, more generally, this thesis.

The setting usually considered in Voting Theory can be formulated as follows. A set $N = \{1, \ldots, n\}$ of *agents*[1] must choose an *alternative* that belongs to the set A, $|A| = m$. The alternatives can be candidates in a political election, but in computational settings the alternatives are often beliefs, joint plans, recommendations, or other conceivable issues. Each agent's preferences are formulated as a linear order (a ranking) over the alternatives. The common choice is determined by a *social choice function* (SCF), which is a function from the preferences of the agents to alternatives.

An ubiquitous SCF, used in nearly all political elections, is the *Plurality* SCF. Under Plurality, each agent awards one point to its top-ranked alternative; the alternative with the largest number of total points wins the election. A less obvious example is the Copeland function, where the winner is the alternative that dominates the most alternatives in pairwise elections; $a \in A$ is said to beat $b \in A$ in a pairwise election if a majority of agents prefer a to b. The formal definitions and notations are given in Chapter 2.

Similarly to Computational Social Choice in general, Computational Voting Theory is an interdisciplinary field where Economics and Computer Science interact. The interaction is mutual, namely it works both ways:

1. Economics applied to Computer Science: applications of Voting Theory techniques to decision making in AI.

2. Computer Science applied to Economics: computational analysis of Voting theory paradigms sheds new light on much studied issues.

Currently the body of work regarding the first item is not large, and yet includes works in areas as diverse as Planning [45], Scheduling [64], Recommender Systems [59], Collaborative Filtering [109], Information Extraction [139], and Computational Linguistics [106]. However, most research on Computational Voting Theory has concentrated on the second item above, and this is indeed our focus here. We presently discuss in slightly more detail some of the major, specific issues (not necessarily the ones featured in this thesis); by no means do we give a full coverage.

1.1.1 Circumventing the G-S Theorem on Computational Grounds

One of the major issues in in Social Choice Theory, which lies at the heart of its intersection with Game Theory, is the problem of manipulation in voting. Recall that an SCF, given the preferences of the agents, returns a winning alternative. However, the truthful preferences of the agents are their private information; the SCF can only rely on the preferences *reported* by the agents. It is self evident that in many settings, agents can benefit by reporting false preferences, that is, may improve the outcome of the election by lying. The quality of the outcome is measured, naturally,

[1] Agents are often referred to as *voters*, or, in some contexts, *players*.

according to the truthful preferences. An agent that reveals its preferences strategically is said to *manipulate* the election. An SCF under which agents can never benefit from manipulation is called *strategyproof*.

The seminal result of Gibbard [60] and Satterthwaite [135] essentially states that manipulation is inescapable. In more detail, the theorem asserts that any SCF that satisfies minimal assumptions is not strategyproof. Since the 1970's, an incalculable amount of work has been devoted to circumventing the Gibbard-Satterthwaite (G-S) theorem. In particular, *Mechanism Design* is a field that can be seen as stemming from ruinous implications of the theorem. The underlying assumption that allows for possibility results is that agents can be compensated by transferring money, thus aligning their incentives with those of the designer. The works of Vickrey [146], Clarke [25], and Groves [62] have laid the foundations of the field of Mechanism Design by introducing the all-important VCG mechanism. For an excellent overview of Mechanism Design, see Nisan [104].

A different path to circumventing the G-S Theorem was introduced in the influential work of Bartholdi et al. [8]. These authors have suggested that the impossibility result can be avoided on computational grounds. Indeed, the agents under consideration, and in particular agents in political elections, can be assumed to be bounded-rational. Thus, even though revealing false preferences in a beneficial way is theoretically possible, it might prove to be a computationally difficult task under certain SCFs. To be more precise, the computational problem is formulated as follows: under a fixed SCF, we are given the preferences of the truthful agents and a preferred alternative p, and asked whether a manipulator can cast a ballot such that p wins. The agenda is therefore to find (among the existing SCFs) or design SCFs that are computationally hard to manipulate.

Bartholdi et al. supported their approach by presenting a specific SCF—Copeland with second order tie breaking—that is $\mathcal{NP}$-hard to manipulate. Decisive evidence to support the approach was ultimately presented by Bartholdi and Orlin [7], who proved that the Single Transferable Vote (STV) is hard to manipulate. STV is one of the prominent SCFs in the literature on voting. It proceeds in rounds; in the first round, each agent votes for the alternative that it ranks first. In every subsequent round, the alternative with the least number of votes is eliminated, and the votes of agents who voted for that alternative are transferred to the next surviving alternative in their ranking (see Chapter 2 for a formal definition).

Two decades later, the agenda suggested by Bartholdi et al. is still the object of significant, and growing, interest. An important step forward was taken by Conitzer and Sandholm [30], who noticed that hardness of manipulation can be induced by tweaking common SCFs, that is by adding a preround. In the preround, the alternatives are paired; the alternatives in each pair compete against each other. The introduction of a preround can make an election $\mathcal{NP}$-hard, $\#\mathcal{P}$-hard, or $\mathcal{PSPACE}$-hard, depending on whether the preround precedes, comes after, or is interleaved with the SCF, respectively. Elkind and Lipmaa [43] generalized this approach using Hybrid SCFs, which are composed of several base SCFs.

Some authors have also considered a setting where there is an entire coalition of manipulators. In this setting, the standard formulation of the manipulation problem is as follows: we are given a set of votes that have been cast, and a set of manipulators. In addition, all votes are weighted, e.g., a agent with weight k counts as k agents voting identically. We are asked whether the manipulators can cast their vote in a way that makes a specific alternative win the election.

Conitzer et al. [35] have shown that the coalitional manipulation problem is $\mathcal{NP}$-hard in a variety of SCFs. Indeed, in this setting the manipulators must coordinate their strategies, on top of taking the weights into account, so manipulation is made much more complicated. In fact, the

problem is so complicated that the hardness results hold even when the number of alternatives is constant. Hemaspaandra and Hemaspaandra [65] generalized some of these last results by exactly characterizing the *scoring functions* (see Chapter 2) in which manipulation is $\mathcal{NP}$-hard. Elkind and Lipmaa [44] have shown how to use cryptographic techniques, namely one-way functions, to make coalitional manipulation hard.

More recently researchers have begun looking at the unweighted version of the coalitional manipulation problem. Faliszewski et al. [50] demonstrated that this version of the problem is still hard under Copeland (under some assumptions on tie breaking). Zuckerman, Procaccia and Rosenschein [155] have established, as corollaries of their main theorems, that the problem is tractable under several prominent SCFs, and gave approximation algorithms for an optimization version of the problem ("How many manipulators are needed in order to make a given alternative win?") under the important Borda and Maximin SCFs. The Borda function is defined as follows: each agent awards $m-1$ points to the alternative it ranks first, $m-2$ points to the second place, etc. The alternative that accumulates the most points wins the election. For a definition of Maximin, see Chapter 2.

Recently researchers have investigated the complexity of manipulation in elections with multiple winners. In general, the assumption is that the manipulator has a utility function on the alternatives, and the question is whether it can cast its vote in a way that guarantees that the total utility of the set of winners be above a given threshold. Procaccia et al. [125] characterized the computational complexity of the multi-winner manipulation problem under several prominent SCFs. Meir, Procaccia and Rosenschein [96] have extended the results of Procaccia et al. [125] by asking whether the above characterization still holds when the manipulator has a more restricted goal in mind, such as including some alternative among the set of winners.

Despite the abundance of results regarding the worst-case complexity of manipulation, some researchers have suggested that worst-case complexity may not be a good enough barrier against manipulation. Indeed, one would ideally like to design a SCF that is hard to manipulate according to some average-case flavor of hardness, that is, computationally hard with respect to almost all instances of the manipulation problem. Several recent works suggest that common SCFs that are in fact hard to manipulate in the worst-case do not satisfy this criterion. We elaborate below.

Procaccia and Rosenschein [119] attempted to establish a framework that would enable showing that manipulation is frequently tractable. They introduced the paradigm of *Junta distributions*, exceptionally hard distributions over the instances of the coalitional manipulation problem. Using their notions, they demonstrated that the family of *scoring functions*, which includes Plurality and Borda, is frequently easy to manipulate when the number of alternatives is constant. The concept of Junta distributions was further discussed at length by Erdélyi et al. [46]. Zuckerman, Procaccia and Rosenschein [155] took a step forward by adopting the general approach of Procaccia and Rosenschein, but refining their results by characterizing the *windows of error* of different manipulation algorithms, i.e. instances on which the algorithms err. Their results, formulated for the coalitional manipulation problem, are conceptually close to approximation results, and in fact directly yield approximation algorithms for the unweighted setting, as mentioned above.

Procaccia and Rosenschein [120] have reconsidered the coalitional manipulation setting. They asked what the relation between the number of manipulators and probability of manipulation is, and found that the threshold is the square root of the number of agents. Specifically, if the number of manipulators is asymptotically smaller than the threshold then the probability is negligible, whereas if it is larger then the probability is almost 1. These results were generalized by Xia and

Conitzer [147].

Another interesting approach was advocated by Conitzer and Sandholm [33], who noticed that SCFs can be frequently manipulated if they satisfy two properties. The first property is quite natural (albeit not satisfied by some common SCFs), whereas the second property is nonintuitive. The authors validated their approach by empirically demonstrating that the second property holds with high probability with respect to most prominent SCFs.

Friedgut et al. [57] have proposed yet another approach to the question of frequency of manipulation. They have shown that if a manipulator simply reports random preferences, it benefits with nonnegligible probability when compared with submitting its true preferences. Hence, drawing a polynomial number of random rankings and submitting the best one yields a beneficial manipulation with high probability. Their results hold under any reasonable SCF, but only when the number of alternatives is exactly three. Xia and Conitzer [148] complemented this result by showing that a similar result holds for any constant number of alternatives, but under stricter assumptions on the SCF.

All the results given above relate to the approach first proposed by Bartholdi et al. [8] for circumventing Gibbard-Satterthwaite on the grounds of computational complexity. We presently briefly discuss a new approach recently introduced by Peleg and Procaccia [107]. They suggested that truthfulness can be induced by assuming the presence of a mediator, and tweaking the solution concept under consideration. More precisely, Peleg and Procaccia have shown how to design SCFs such that, given the existence of a mediator, even coalitions of agents cannot benefit by lying. Peleg and Procaccia [108] later extended this investigation to a characterization of social choice correspondences (functions from preferences to sets of alternatives) where truth-telling is in equilibrium, assuming a mediator.

1.1.2 Control and Bribery

While manipulation, and circumventing the Gibbard-Satterthwaite Theorem, might well be the single most important issue in Computational Voting Theory, closely related issues have also received much attention. Another seminal paper by Bartholdi et al. [10] introduced the problem of *control* in elections. In the basic setup, the authority in charge of the election—known as the *chairman*—seeks to influence its outcome by tampering with the set of registered agents or the set of available alternatives. For instance, the chairman can add agents that support some cause, or remove strong alternatives that might cause a favorite alternative to lose. Bartholdi et al. studied the complexity of seven different types of control under two SCFs. The authors reached the conclusion that different SCFs differ significantly in terms of their resistance to control. Hemaspaandra et al. [67] extended these results to the destructive setting, where the chairman wishes for a specific alternative to lose the election rather than win it.

Hemaspaandra et al. [68] asked whether it is possible to design a SCF that is fully computationally resistant to control. They showed that there is an SCF, obtained as a hybrid of other functions, which is resistant to twenty different types of control. Some common SCFs were later shown to come close to this ideal of total resistance to control [49, 51].

Faliszewski et al. [48] introduced a variation on the control setting: the *bribery* problem. Here, the chairman must bribe agents in order to win them over. The authors give a characterization of the complexity of bribery under several SCFs. Faliszewski [47] extended this setup by defining and characterizing the complexity of the *nonuniform bribery* problem, where the corrupt agents' prices depend on the exact nature of the change in their votes that is requested.

1.1.3 Winner Determination

We turn to yet another important agenda introduced by Bartholdi et al. [9]. They suggested that, under some SCFs, determining the winner of the election may be a hard computational problem. Note that, in stark contrast to the problems discussed above, in this case computational complexity is a negative phenomenon rather than a positive one, as it may prevent the SCF from being used in practice.

Bartholdi et al. demonstrated that, under the interesting function proposed by Charles Dodgson in the 19th century, determining the winner of the election is $\mathcal{NP}$-hard. Under Dodgson's function, an alternative's score is the number of exchanges between adjacent alternatives in the agents' preferences that must be performed in order to make that alternative beat every other alternative in pairwise elections. An exact characterization of the complexity of this problem remained elusive until Hemaspaandra et al. [66] proved that it is complete for the complexity class Θ_2^p. Rothe et al. [132] subsequently showed that winner determination under the closely related SCF proposed by Young [154] is also complete for Θ_2^p. Procaccia et al. [123] designed an algorithm that approximates an alternative's score under Dodgson's function to a factor of $\mathcal{O}(\log m)$, but proved that Young's function is hard to approximate by any factor.

Similarly, the related social welfare functions (that map the preferences of the agents to rankings over alternatives) proposed by Kemeny and Slater have been shown to be Θ_2^p-complete to decide [9, 2]. Kemeny's function aggregates the rankings of the agents into a ranking that minimizes the total sum of disagreements, over pairs of alternatives, with the individual rankings. Slater's function chooses the ranking that most agrees with the *majority* of agents regarding pairs of alternatives. Davenport and Kalagnanam [38], and later Conitzer et al. [34], provided heuristic algorithms for exactly computing the results of an election under Kemeny's function, while Ailon et al. [1] designed approximation algorithms for Kemeny. Heuristic algorithms for computing the results of Slater's function have also been the subject of interest [72, 26].

Procaccia et al. [127] discuss the complexity of winner determination under the prominent social choice correspondences proposed by Monroe [99] and by Chamberlin and Courant [22]. These two correspondences basically elect a set of alternatives that minimizes the total *misrepresentation* of the agents; the goal is to achieve fully proportional representation: a faction of agents should be represented in the elected set of alternatives in a way that is proportional to its size. Procacia et al. show that winner determination is $\mathcal{NP}$-hard in both schemes, but the problem is tractable when the number of alternatives to be elected is constant.

Slightly further afield, some recent work explored the complexity of computing tournament choice sets [17, 18, 19, 71]. A tournament is a complete asymmetric relation on the set of alternatives; a tournament is often used to model the results of all possible pairwise elections between pairs of alternatives. Tournament choice sets single out sets of "best" alternatives in a tournament, according to different criteria. The works mentioned above, put together, give a complete characterization of the computational complexity of most prominent choice sets.

1.1.4 Vote Elicitation

Despite the results outlined in the previous subsection, elections held under most prominent SCFs are easy to decide. Nevertheless, in plausible settings, especially those where communication is restricted or error-prone, one may be interested in obtaining as little information as possible from the agents in a way that is sufficient to determine the outcome of the election. This is known as

vote elicitation.

Conitzer and Sandholm [28] defined several computational problems related to vote elicitation. For instance, in the effective elicitation problem the question is whether there is a small subset of agents that can decide the outcome of the election. Conitzer and Sandholm, *inter alia*, showed this problem to be $\mathcal{NP}$-hard under several SCFs.

Another way to approach the vote elicitation setting is to assume that the agents only submit incomplete preferences, i.e. for a given agent, its ordering over alternatives is not necessarily complete. An alternative is a *possible* winner if it wins for some completion of the preferences, and a *necessary winner* if it wins under all completions. Characterizations of the complexity of determining possible and necessary winners appear in several works [81, 111, 149].

In the communication complexity model, we are only interested in the number of bits transferred between the agents, that is the amount of information sent and received. This concrete complexity model is perhaps even more appropriate, in the context of vote elicitation, than computational complexity. Conitzer and Sandholm [32] demonstrated that, while some SCFs require very little information, others practically need an amount of information asymptotically equivalent to the entire preference profile of the agents. In closely related work, Segal [137] characterized the communication complexity of a large class of SCFs. Conitzer [27] investigated the problem in the query complexity model, and under the assumption that agents have single peaked preferences. As a canonical example for single peaked preferences, consider a setting where the alternatives are points on the real line; each agent has an ideal *bliss point*, and the closer a point is to the bliss point the more preferred it is.

1.1.5 Combinatorial Voting

In many domains, in particular those that arise in AI, the preferences of the agents have a combinatorial structure. Specifically, if the agents are voting on multiple issues, their preferences over the issues can be interdependent. This significantly increases the computational complexity of SCFs [84].

An intriguing approach is to try to decompose the social choice function into votes on individual issues. A barrier that must be overcome is the phenomenon known as *multiple election paradoxes* [16]. For example, suppose there are two boolean-valued issues Y and Z; 10 agents want Y but don't want Z ($Y\bar{Z}$), 10 agents want Z and not Y ($\bar{Y}Z$), and one agent wants both Y and Z (YZ). Voting separately on the two issues would lead to the outcome YZ, even though this outcome is preferred only by one agent.

Well known SCFs, such as Borda, cannot be decomposed [85]. Nevertheless, recent papers give sufficient conditions and techniques for designing decomposable SCFs [150, 151].

1.2 Structure and Overview of Results

Chapter 2 of the thesis gives an introduction to Voting Theory. In particular, we present the basic concepts and notations and introduce the prominent social choice functions. We then discuss tournaments and voting trees. Finally, we formulate the Gibbard-Satterthwaite Theorem [60, 135].

The bulk of the thesis is devoted to the presentation of our results. The presentation consists of three parts, where each part contains two chapters. We elaborate below on the structure of this partition and the results given therein.

Part I: Elections and Approximation

Approximation algorithms are one of the major areas of research in the modern theory of algorithms. Usually the goal is to solve a computationally intractable optimization problem in a manner which is computationally efficient, albeit only approximate. Specifically, we say that an algorithm is an α-approximation algorithm if the quality of its solution is always (in the *worst-case*) worse than the optimal solution by at most a factor of α. In Part I we deal with approximation algorithms in the traditions sense, but also find a novel application for the concept of approximation.

Chapter 3: Approximability of Dodgson and Young Elections. Some previous work has dealt with approximating social welfare functions that are hard to resolve [1, 36, 77]. We continue this line of work by studying the approximability of two prominent SCFs: Dodgson and Young.

Charles Dodgson (better known by his pen name, Lewis Carroll) suggested an appealing voting system in 1876. Unfortunately, at the time Dodgson did not take into account such futuristic considerations as computational complexity, and, as it turned out more than a century later, computing the Dodgson score of an alternative is NP-hard [9].

In order to understand the SCF suggested by Dodgson, we must go even further back in time. The French mathematician Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet, suggested (as early as the 18th century) the following criterion for resolving an election: choose an alternative that is preferred to any other alternative by a majority of agents. However, the marquis himself noticed that such an alternative, known as a *Condorcet winner*, does not always exist.

Dodgson suggested to choose the alternative closest to being a Condorcet winner. Specifically, the *Dodgson score* of an alternative is the minimum number of exchanges that must be introduced in the preferences of the agents in order to make said alternative a Condorcet winner. Young [154] followed the same line of reasoning; the *Young score* of an alternative is the size of the maximum subset of agents for which the alternative is a Condorcet winner. The Young score is also hard to compute [132].

Our results are two-fold. In the context of approximating the Dodgson score, we devise an $\mathcal{O}(\log m)$ randomized approximation algorithm, where m is the number of alternatives. Our algorithm is based on solving the linear program proposed by Bartholdi et al. [9] and using randomized rounding. It follows from a result of McCabe-Dansted [92] that no polynomial-time randomized algorithm can approximate the Dodgson score to within an expected ratio of $\Omega(\log m)$ (unless $\mathcal{NP} = \mathcal{RP}$), so this result is asymptotically optimal.

The problem of calculating the Young score seems simpler at first glance. Therefore, our result with respect to this problem is quite surprising: it is $\mathcal{NP}$-hard to approximate the Young score by any factor. Specifically, we show that it is $\mathcal{NP}$-hard to distinguish between the case where the Young score of a given alternative is 0, and the case where the score is greater than 0.

Chapter 4: Approximating Maximum Degree in a Tournament by Binary Trees. A tournament is a complete and asymmetric (dominance) relation over a set of alternatives. Tournaments appear in many contexts but are closely linked to voting theory, since the dominance relation is often used to represent the preferences of the majority, that is, alternative a dominates b if the majority of agents prefer a to b.

A voting tree is a binary tree whose leaves are labeled by alternatives; such trees describe an iterative procedure for choosing a winning alternative from a tournament. At each stage, two

sibling leaves compete, the winner according to the given tournament survives and proceeds to the father. The alternative that reaches the root in this way is the winner.

Previous work in economics [52, 95, 98, 102, 69, 41, 143, 37] has investigated which functions from tournaments to alternatives can be realized by voting trees. In particular, it is known that there is no voting tree such that, given any tournament, always chooses a Copeland winner. To elaborate a bit, the *Copeland score* of an alternative in a tournament is the number of other alternatives beaten by this alternative. A *Copeland winner* is an alternative that maximizes the Copeland score.

We apply the Computer Science-oriented concept of approximation to this setting. Indeed, we ask whether there exist voting trees that always choose alternatives with Copeland score that approximates the score of the winner. We investigate this question in two models: a deterministic model, and a randomized model that allows arbitrary distributions over trees, and considers the expected score of the winner.

Our main negative results are upper bounds of 3/4 and 5/6, respectively, on the approximation ratio achievable by deterministic trees and randomizations over trees. We find it quite surprising that randomizations over trees cannot achieve a ratio arbitrarily close to 1.

For most of the chapter we concentrate on the randomized model. We study a class of trees we call voting caterpillars, which are characterized by the fact that they have exactly two nodes on each level below the root. We devise a randomization over "small" trees of this type, which further satisfies an important property we call *admissibility*: its support only contains trees where every alternative appears in some leaf. Our main positive result is the construction of an admissible randomization over voting trees of size polynomial in m with an approximation ratio of $1/2 - \mathcal{O}(1/m)$. We prove this theorem by establishing a connection to a nonreversible, rapidly mixing random walk on the tournament, and analyzing its stationary distribution. The proof of rapid mixing involves reversibilizing the transition matrix, and then bounding its spectral gap via its conductance. To the best of our knowledge, this constitutes the first use of rapid mixing, and in particular of notions like conductance, as a proof technique in Computational Economics. We further show that our analysis is tight, and that voting caterpillars also provide a lower bound of 1/2 for the second order degree of an alternative, defined as the sum of degrees of those alternatives it dominates.

The chapter concludes with negative results about more complex tree structures, which turn out to be rather surprising. In particular, we show that the approximation ratio provided by randomized balanced trees can become arbitrarily bad with growing height. We further show that "higher-order" caterpillars, with labels chosen by lower-order caterpillars instead of uniformly at random, can also cause the approximation ratio to deteriorate.

Part II: Elections and Computational Learning

Broadly speaking, computational learning theory tackles the following problem. Given sample values for an unknown target function, find a function that is generally "close" to the target function. The target function is often assumed to belong to some fixed function class, hence it is possible to determine how many samples are needed to achieve a good generalization based on the combinatorial properties of the function class.

In Part II of the thesis, we deal with the interplay between voting theory and computational learning, but in two opposite directions: one chapter deals with the application of learning theory to the design of SCFs, whereas the other deals with the application of voting and mechanism design

paradigms to improve the machine learning process itself. The latter chapter also ties in nicely to our results regarding approximation (given in Part I), as a substantial part of the chapter studies approximation in a mechanism design setting without payments.

Chapter 5: The Learnability of Social Choice Functions. SCFs can be regarded as functions to be learned in a machine learning model. The input space is the space of all possible preference profiles, while the output space is the set of alternatives. In this setting, it is natural to investigate the complexity, both in the computational sense and in the learning-theoretic sense, of learning prominent classes of SCFs.

We motivate this agenda by relating it to the question of designing SCFs. Think of a designer who has in mind some SCF; this function can be inefficiently represented, e.g., by a huge table that lists all the possible preference profiles and the corresponding winners. So, the goal is to design an SCF that is concisely representable and close to what the designer has in mind, while asking the designer as few queries as possible and investing as little computational effort as possible. We investigate these questions in the context of two prominent families of SCFs: scoring functions and voting trees.

A scoring function can be represented by a vector of real numbers $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle$. We show that scoring functions are efficiently learnable, that is, it is possible to learn a scoring function "close" to the target function in time polynomial in the number of agents and alternatives; in particular, the number of queries to the designer is also polynomial. We achieve this result by giving bounds on the *generalized dimension* of the class of scoring functions, a measure of the combinatorial richness of this class.

Next, we address the class of voting trees. We show that in general, in order to learn an SCF close to a target voting tree, an exponential number of queries is needed. However, the goal can be achieved with a polynomial number of queries if the target voting tree has a polynomial number of leaves. We further study the computational aspects of the problem, showing that a related decision problem is $\mathcal{NP}$-hard, but providing experimental data that suggests that the problem can be solved in practice for reasonable instances.

Finally, we ask whether it is possible to extend this approach. Specifically, we pose the question: given a class of SCFs, if the designer has some general SCF in mind (rather than an SCF that is known to belong to this class), is it possible to learn a "close" rule from this class? We answer this question in the negative with respect to our two classes of SCFs.

Chapter 6: Strategyproof Regression Learning. Regression learning deals with learning real-valued functions. The accuracy of the learning process is measured according to a *loss function*, which measures the distance between the values of the target function and the function returned by the learner. Common examples of loss functions are the *squared loss*, which returns the square of the Euclidean distance, and the *absolute loss*, which is simply the Euclidean distance.

In our setting we have, in addition, a set of strategic agents. Each agent holds as private information a distribution over the input space, which reflects the relative importance it gives to different issues, as well as its own values for the points of the input space. The cost of each agent is given by the expected distance between the function returned by the learner and the agent's own values, weighted by the distribution of the agent. The designer's goal is to minimize the total cost of the agents. The examples that are used in the learning process are elicited from the agents by sampling their distributions; the agents might lie about the values of the sampled examples in

order to sway the outcome of the learning process to one they find more favorable.

Before elaborating on our results, we briefly touch on the relation between this work and voting theory. Ultimately, we shall see that the foregoing setting reduces to an interesting mechanism design setting that does not involve sampling. In the latter setting, it is possible to obtain strategyproofness results even without payments, by leveraging a significant body of research from voting theory. Hence, although at first glance this chapter may seem unrelated to voting, in fact the two are intimately connected.

We begin our investigation by considering a restricted setting where each agent is only interested in a single point of the input space. Quite surprisingly, it turns out that a specific choice of loss function, namely the absolute loss function, leads to excellent game-theoretic properties: an algorithm which simply finds an empirical risk minimizer on the training set is group strategyproof, meaning that no coalition of agents is motivated to lie. We also show that even much weaker truthfulness results cannot be obtained for a wide range of other loss functions, including the popular squared loss.

In the more general case where agents are interested in non-degenerate distributions, achieving incentive compatibility requires more sophisticated mechanisms. We show that the well-known VCG mechanism does very well: with probability $1 - \delta$, no agent can gain more than ϵ by lying, where both ϵ and δ can be made arbitrarily small by increasing the size of the training set. This result holds for any choice of loss function.

We also study what happens when payments are disallowed. In this setting, we obtain limited positive results for the absolute loss function and for restricted yet interesting function classes. In particular, we present a mechanism which is approximately group strategyproof as above and 3-efficient in the sense that the solution provides a 3-approximation to optimal social welfare. We complement these results with a matching lower bound and provide strong evidence that no approximately incentive compatible and approximately efficient mechanism exists for more expressive function classes.

Part III: Frequency of Manipulation in Elections

Part III of the thesis presents two approaches to dealing with the question: is manipulation in elections frequently hard under typical distributions on the preferences of the agents? An algorithmic approach is presented in Chapter 7, and a descriptive approach is given in Chapter 8. It is important to note that both chapters deal with manipulation by coalitions (the coalitional manipulation problem) rather than by individual manipulators; the former problem is computationally much harder than the latter. Since some general background was already given in Section 1.1.1, in the sequel we simply describe our approaches and state our results.

Chapter 7: Junta Distributions. Our goal in this chapter is to show that manipulation might be tractable under typical distributions, even under SCFs that are known to be hard to manipulate in the worst-case. The greatest obstacle is coming up with an "interesting" distribution of preference profiles with respect to which the complexity is computed, and our solution may be controversial. We analyze manipulation problems that are distributed with respect to a *Junta distribution*. Such a distribution must satisfy several conditions, which (arguably) guarantee that it focuses on preference profiles that are harder to manipulate. We consider an SCF to be susceptible to manipulation when there is a polynomial time algorithm that can usually manipulate it: the probability of failure (when

the instances are distributed according to a Junta distribution) must be inverse-polynomial. Such an algorithm is known as a *heuristic* polynomial time algorithm.

We then show that the family of scoring functions, mentioned several times above, can be frequently manipulated, even when the preference profiles are distributed according to a Junta distribution, if the number of alternatives is constant. Specifically, we contemplate *sensitive* scoring functions, which include such well-known functions as Borda and Veto. To accomplish this task, we define a natural distribution μ^* over the instances of a well-defined coalitional manipulation problem, and show that this is a Junta distribution. Furthermore, we present the manipulation algorithm Greedy, and prove that it usually succeeds with respect to μ^*. The significance of this result stems from the fact that sensitive scoring functions are $\mathcal{NP}$-hard to manipulate, *even* when the number of alternatives is constant. We support our claim that Junta distributions provide a good benchmark by proving that Greedy also usually succeeds with respect to the uniform distribution.

Chapter 8: The Fraction of Manipulators The last results that are included in the thesis deal with the probability that a coalition of manipulators has the power to sway the outcome of the election. Intuitively, if the size of the coalition is small then this probability is small, under preferences that are reasonably distributed. If the coalition is very large, then the probability must be close to 1. In other words, it is either almost always possible to find a successful manipulation, or almost never possible. Chapter 8 makes this intuition more accurate.

We notice that the correct option (small or large probability) depends only on easily testable properties of the distribution, and on the fraction of manipulators. If n is the number of agents and $\hat{n}$ is the number of manipulators in the coalition, we demonstrate that, when $\hat{n} = o(\sqrt{n})$, manipulation is almost never possible under almost any distribution where the agents vote independently. When $\hat{n} = \omega(\sqrt{n})$, we characterize the distributions where manipulation is almost always possible, and the ones where it is almost never possible. We rigorously prove these results in the context of the family of scoring functions.

Ultimately, our results yield a generic algorithm that usually decides the coalitional manipulation problem under many natural distributions.

1.3 Prerequisites

This thesis requires basic (graduate-level) knowledge of the theory of computer science on the part of the reader. In particular, the reader is assumed to be (at least generally) familiar with the following topics: basic complexity theory, approximation algorithms, linear programming, Markov chains, basic probability theory, basic algebra. A significant portion of the thesis (Part II) deals with learning theory, but the necessary concepts and theorems are introduced in the relevant chapters.

On the other hand, the thesis is completely self-contained with respect to its economic aspects. To put it differently, any graduate student in computer science should be able to read and understand the entire thesis. Passing knowledge of game theory and mechanism design may help understand some of the concepts that are dealt with, but such knowledge is certainly not a prerequisite. Most importantly, no prior knowledge of voting theory is required.

1.4 Bibliographic Notes

Chapter 3 is based on joint work with Michal Feldman and Jeff Rosenschein; a significantly extended version appeared as [21]. Chapter 4 is based on joint work with Felix Fischer and Alex Samorodnitsky [56]. Chapter 5 is based on joint work with Yoni Peleg, Jeff Rosenschein, and Aviv Zohar [128]. Chapter 6 is based on joint work with Ofer Dekel and Felix Fischer [39]. Chapters 7 and 8 are based on joint work with Jeff Rosenschein [119, 120].

1.4.1 Excluded Research

Many topics that I have worked on during my PhD studies have been left out of this thesis, mainly in order to adhere to the Hebrew University's strict page limit for PhD theses. Many of these works lie within the boundaries of computational voting theory, some do not. The excluded research includes (but is not limited to):

Computational Voting Theory

- Work on the distortion of cardinal preferences in voting [116] and the robustness of SCFs [124], which are related to the topics discussed in Part I of this thesis.
- Additional work on frequency of manipulation in elections [155, 40], intimately related to Part III.
- Work on worst-case complexity issues related to elections with multiple winners, both with respect to their strategic aspects [125, 96], and winner determination [127].
- Recent work on Strategyproof learning [97], an extension of the work presented in Chapter 6.
- Extensions of the work on approximating Dodgson and Young elections [21], given in Chapter 3.
- Work on Mediated equilibria, their application in voting, and implementation [107, 108].
- Other works on computational aspects of voting [152, 114].

Other topics

- Work on cooperative games: communication complexity [117], learning [118], and computation of power indices [4].
- Work on argumentation [115].
- Work on reputation systems [122].
- Work on solution concepts for noncooperative games [121].

Chapter 2

Preliminaries

In this chapter we shall formally introduce the mathematical definitions and notations that will serve us throughout this thesis. We may introduce some additional notations later, on an *ad hoc* basis.

2.1 The Basics

We deal with a finite set of *agents* $N = \{1, \ldots, n\}$, and a finite (unless explicitly stated otherwise) set of *alternatives* A, where $|A| = m$. We denote alternatives by letters, usually using a, b, c, x, y, and p. Agent indices usually appear in superscript, whereas alternative indices usually appear in subscript.

Each agent $i \in N$ holds a quasi-order R^i over A, i.e. R^i is a binary relation over A that satisfies reflexivity, antisymmetry, transitivity and totality. Informally, R^i is a *ranking* of the alternatives. The set $\mathcal{L} = \mathcal{L}(A)$ is the set of all such (linear) quasi-orders, so for all $i \in N$, $R^i \in \mathcal{L}$ throughout. A *preference profile* R^N is a vector $\langle R^1, \ldots, R^n \rangle \in L^N$. We sometimes use R^S to denote the preferences of a coalition $S \subseteq N$; xR^Sy means that xR^iy for all $i \in S$.

We are now in a position to define—in one stroke!—three central concepts.

Definition 2.1.1.

1. A *social choice function* (SCF) is a function $f : \mathcal{L}^N \to A$.

2. A *social welfare function* (SWF) is a function $f : \mathcal{L}^N \to \mathcal{L}$.

3. A *social choice correspondence* (SCC) is a function $f : \mathcal{L}^N \to 2^A \setminus \{\emptyset\}$.

Most importantly, an SCF determines the outcome of the election given the preferences of the agents.

2.2 Common SCFs

In the section we describe some prominent SCFs that we shall deal with.

2.2.1 Scoring Functions

The predominant—ubiquitous, even—SCF in political elections is the *Plurality* function. Under Plurality, each agent awards one point to the alternative it ranks first, i.e., its most preferred alternative. The alternative that accumulated the most points, summed over all agents, wins the election. Another example of an SCF is the *Veto* rule: each agent "vetoes" a single alternative; the alternative that was vetoed by the fewest agents wins the election. Yet a third example is the *Borda* rule, devised as early as 1770 by Jean-Charles de Borda: every agent awards $m-1$ points to its top-ranked alternative, $m-2$ points to its second choice, and so forth—the least preferred alternative is not awarded any points. Once again, the alternative with the most points is elected.

The abovementioned three SCFs all belong to an important family of SCFs known as *scoring functions*. A scoring function can be expressed by a vector of parameters $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle$, where each α_l is a real number and $\alpha_1 \geq \cdots \geq \alpha_m$. Each agent awards α_1 points to its most-preferred alternative, α_2 to its second-most-preferred alternative, etc. Naturally, the alternative with the most points wins. Under this unified framework, we can express our three rules as:

- *Plurality*: $\alpha = \langle 1, 0, \ldots, 0 \rangle$.
- *Borda*: $\alpha = \langle m-1, m-2, \ldots, 0 \rangle$.
- *Veto*: $\alpha = \langle 1, \ldots, 1, 0 \rangle$.

Remark 2.2.1. Formally, scoring functions are defined as SCCs, so that all alternatives with maximal score (there may be multiple such alternatives) are elected. In practice, in most cases we will assume some method of tie-breaking in order to obtain SCFs.

Example 2.2.2. Let us present an example to illustrate the differences between different scoring functions. This example is also meant to clarify some of the definitions introduced earlier. Let the set of agents be $N = \{1, 2, 3, 4\}$, and let the set of alternatives be $A = \{a, b, c\}$. Define a preference profile as follows:

R^1	R^2	R^3	R^4
a	c	c	b
b	a	a	a
c	b	b	c

Under Plurality a has one point, b has one, and c has two, thus c is the winner. Under Borda, a has 5 points, b has 3, and c has 4, hence a is the winner. Under Veto, a is again the winner since it was not vetoed by any of the agents.

2.2.2 Single Transferable Vote and Plurality with Runoff

We presently introduce two additional, related, SCFs.

Single Transferable Vote (STV) STV is an SCF that is actually used in political elections around the world. More importantly, different organizations and pressure groups are strongly advocating its use in elections in the United States and United Kingdom.

Under STV, the election proceeds in $m-1$ rounds. In each round, the alternative's score is the number of agents that rank it highest among the remaining alternatives; the alternative with the lowest score is eliminated, and the remaining alternatives advance to the next round.

Plurality with Runoff This SCF is reminiscent of STV, but involves only two rounds. Only two alternatives survive the first round, and proceed to the second. In the second round, the two alternatives that survived the first face off in a *pairwise election*; the winner of the pairwise election between a and b is the alternative that is preferred to the other by a majority of agents.

2.2.3 Condorcet Consistent SCFs

As early as the 18th century the French mathematician and philosopher, Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet, proposed a compelling criterion for selecting the winner of an election. Condorcet proposed that the winner be the alternative that beats every other alternative in a pairwise election. Sadly, it is fairly easy to see that the preferences of the majority may be cyclic, hence a *Condorcet winner* does not necessarily exist. This unfortunate phenomenon is known as the *Condorcet paradox* (see Black [14]).

Given this reality, different SCFs have been devised to satisfy the property known as *Condorcet consistency*: the SCF must elect a Condorcet winner if one exists. In this section we discuss several such functions.

Copeland The Copeland score of an alternative is the number of other alternatives it beats in pairwise elections. Notice that if a Condorcet winner exists, it must have a Copeland score of $m-1$, whereas other alternatives have a score of at most $m-2$ (since they are beaten by the Condorcet winner). Hence, Copeland is Condorcet consistent.

Maximin The Maximin function, also known as *Simpson*, works as follows. For any two alternatives x and y, let

$$N(x,y) = |\{i \in N : \ xR^i y\}|$$

be the number of agents who prefer x to y (given R^N). The Maximin score of x is $\min_{y \neq x} N(x,y)$. In words, the score of an alternative is the result of its worst pairwise election. The winner under Maximin maximizes this minimum, hence the name of the function.

A Condorcet winner must have a Maximin score of more than $n/2$, since it is preferred to any other alternative by a majority of agents. On the other hand, a different alternative loses to the Condorcet winner (if one exists) in a pairwise election, hence its Maximin score is smaller than $n/2$. Therefore, Maximin is Condorcet consistent.

Dodgson and Young Charles Dodgson, better known by his pen name Lewis Caroll, was a mathematician and writer.[1] Dodgson proposed an SCF that chooses the alternative "closest" to being a Condorcet winner. Formally, The Dodgson score of a given alternative x, with respect to a given preference profile R^N, is the least number of exchanges between adjacent alternatives in R^N needed to make x a Condorcet winner.

Example 2.2.3. For instance, let $N = \{1,2,3\}$, $A = \{a,b,c\}$, and let R^N be given by:

R^1	R^2	R^3
a	b	a
b	a	c
c	c	b

[1] Dodgson famously authored "Alice's Adventures in Wonderland".

In this example, the Dodgson score of a is 0 (a is a Condorcet winner), b's score is 1, and c's is 3.

Young [154] raised a second option: measuring the distance by agents. The *Young score* of x with respect to R^N is the size of the largest subset of agents such that x is a Condorcet winner with respect to these agents. If for every nonempty subset of agents x is not a Condorcet winner, then its Young score is 0. In the profile given in Example 2.2.3, the Young score of a is 3, the score of b is 1, and the score of c is 0.

2.3 Tournaments and Voting Trees

A *tournament* T on A is an orientation of the complete graph with vertex set A. In other words, T is a complete and asymmetric relation over A. For a tournament $T \in \mathcal{T}(A)$, we write aTb if the edge between a pair $a, b \in A$ of alternatives is directed from a to b, or a *dominates* b. We denote by $\mathcal{T}(A)$ the set of all tournaments on A.

In voting theory a tournament T is often used to represent the results of all possible pairwise elections given a profile, where aTb means that a beats b in a pairwise election. The following seminal theorem gives an important relation between preference profiles and tournaments.

Theorem 2.3.1 (McGarvey [94])**.** *Let A be a set of alternatives. For every tournament T on A there exists a set of agents N and a preference profile R^N that induces T.*

Notice that the Copeland rule takes into account only the tournament induced by R^N, and essentially elects an alternative with maximum degree in the tournament. Another important class of functions from $\mathcal{T}(A)$ to A is known as *voting trees*. Informally, a voting tree over A is a binary tree with leaves labeled by elements of A. Given a tournament T, a labeling for the internal nodes is defined recursively by labeling a node by the label of its child that beats the other child according to T (or by the unique label of its children if both have the same label). The label at the root is then deemed the winner of the voting tree given tournament T. This definition expressly allows an alternative to appear multiple times in the leaves of a tree.

For example, assume that the alternatives are a, b and c, and bTa, cTb and aTc. In the tree given in Figure 2.1, b beats a and is subsequently beaten by c in the right subtree, while a beats c in the left subtree. a and c ultimately compete at the root, making a the winner of the election.

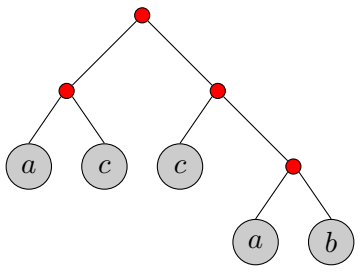


Figure 2.1: An example voting tree.

Formally, a *voting tree* on A is a structure $\Gamma = (V, E, \ell)$ where (V, E) is a binary tree with root $r \in V$, and $\ell : V \to A$ is a mapping that assigns an element of A to each leaf of (V, E). Given a

tournament T, a unique function $\ell_T : V \to A$ exists such that

$$\ell_T(v) = \begin{cases} \ell(v) & \text{if } v \text{ is a leaf} \\ \ell(u_1) & \text{if } v \text{ has children } u_1 \text{ and } u_2, \text{ and } \ell(u_1) T \ell(u_2) \text{ or } \ell(u_1) = \ell(u_2) \end{cases}$$

We are interested in the label of the root r under this labeling, which we call the winner of the tree and denote by $\Gamma(T) = \ell_T(r)$.

2.4 Manipulation and the G-S Theorem

For the first time in this chapter, we differentiate between two layers in the agents' preferences: their truthful preferences, which are the private information of the agents, and their reported preferences, which are used as input to the SCF and therefore affect the social outcome. Once this distinction is made, there is cause for concern since agents may report untruthful preferences in an attempt to improve the outcome of the election; this phenomenon is known as *manipulation*. If multiple agents try their hand at manipulating the election at the same time, the chosen alternative may be one that is far from being socially desirable.

Definition 2.4.1. Let $f : \mathcal{L}^N \to A$ be an SCF. f is *strategyproof* if for all $R^N \in \mathcal{L}^N$, all agents $i \in N$ and all $Q^i \in \mathcal{L}$, $f(R^N) R^i f(Q^i, R^{N\setminus\{i\}})$, where $(Q^i, R^{N\setminus\{i\}})$ is identical to R^N except for the replacement of R^i by Q^i.

In words, f is strategyproof if for any preference profile R^N, every agent i prefers (according to R^i) the outcome when it reports its true preferences at least as much as the outcome resulting from the report of any different ranking Q^i. It is implicitly assumed here that a potential manipulator has complete information about the ballots of the other agents, namely $R^{N\setminus\{i\}}$. This is essentially a worst-case assumption: we would like the SCF to be strategyproof *even* if the manipulator has complete information.

We say that an SCF f is *dictatorial* if there exists a *dictator* $d \in N$ such that for all R^N, $f(R^N)$ is the alternative ranked first in R^d. f is said to be nondictatorial if there is no such dictator. The famous Gibbard-Satterthwaite (G-S) Theorem asserts that, essentially, there is no SCF that is both strategyproof and nondictatorial.

Theorem 2.4.2 (Gibbard-Satterthwaite [60, 135])**.** *Let $f : L^N \to A$ be an SCF onto A, $|A| \geq 3$. If f is strategyproof, then f is dictatorial.*

Part I

Elections and Approximation

Chapter 3

Approximability of Dodgson and Young Elections

3.1 Introduction

One of the big questions in social choice theory is: given the preferences of the agents, which alternative best reflects the social good? As mentioned in Section 2.2, the Marquis de Condorcet suggested the following intuitive criterion: the winner should be an alternative that beats every other alternative in a *pairwise election*, i.e., an alternative that is preferred to any other alternative by a majority of the agents. However, a Condorcet winner might not always exist.

In order to circumvent this situation, several researchers have proposed choosing an alternative that is "as close as possible" to a Condorcet winner. Different notions of proximity can be considered, and yield different SCFs. Two of these notions were presented in Section 2.2: Dodgson's rule measures the distance according to the number of exchanges between adjacent alternatives, whereas Young's rule measures the distance by agents.

Though these two SCFs sound appealing and straightforward, they are notoriously complicated to resolve. As early as 1989, Bartholdi, Tovey and Trick [9] have shown that computing the Dodgson score is $\mathcal{NP}$-complete, and that pinpointing a Dodgson winner is $\mathcal{NP}$-hard. This important paper was one of the first to introduce complexity-theoretic considerations to social choice theory. Hemaspaandra et al. [66] refined the abovementioned result by showing that the Dodgson winner problem is complete for Θ_2^p, the class of problems that can be solved by $\mathcal{O}(\log n)$ queries to an $\mathcal{NP}$ set. Subsequently, Rothe et al. [132] proved that the Young winner problem is also complete for Θ_2^p.

The abovementioned complexity results give rise to the agenda of *approximately* calculating an alternative's score, under the Dodgson and Young schemes. This is clearly an interesting computational problem, as an application area of algorithmic techniques.

However, from the point of view of social choice theory, it is not immediately apparent that an approximation of a SCF is satisfactory, since an "incorrect" alternative—in our case, one that is not closest to a Condorcet winner—can be elected. Nevertheless, we argue that the use of such an approximation is strongly motivated. Indeed, at least in the case of the Dodgson and Young rules, the winner is an "approximation" in the first place, in instances where no Condorcet winner exists. Moreover, the approximation algorithm is equivalent to a new SCF, which is guaranteed to elect an alternative that is not far from being a Condorcet winner. In other words, a perfectly sensible

definition of a "socially good" winner, given the circumstances, is simply the alternative chosen by the approximation algorithm. Note that the approximation algorithm can be designed to satisfy the Condorcet criterion, i.e., always elect a Condorcet winner if one exists (this is always true for an approximation of the Dodgson score, as the score of a Condorcet winner is 0, and is indeed the case here).

3.2 Approximability of Dodgson

In this section, we present the main result of the chapter: an LP-based randomized rounding algorithm that gives an $\mathcal{O}(\log m)$ approximation for the Dodgson score of an alternative. Let us first introduce some notation. Let $a^* \in A$ be a distinguished alternative, whose Dodgson score we wish to compute. Define the *deficit* of a^* with respect to $a \in A$, simply denoted $\mathrm{def}(a)$ when the identity of a^* is clear, as the number of additional agents that must rank a^* above a in order for a^* to beat a in a pairwise election. For instance, if 4 agents prefer a to a^* and only one agent prefers a^* to a, then $\mathrm{def}(a) = 2$. If a^* beats a in a pairwise election (namely a^* is preferred by the majority of agents) then $\mathrm{def}(a) = 0$.

As a warm-up, we start by considering some trivial combinatorial algorithms. Recall that in order to compute the Dodgson score of a given alternative under some preference profile, we must perform the minimal number of exchanges between adjacent alternatives. In fact, clearly the only type of exchanges to be considered are the ones that move the given alternative upward in some ranking, at the expense of some other alternative. In other words, we can simply talk about the number of positions each agent pushes the given alternative.

An approximation algorithm that immediately comes to mind is the following greedy algorithm.

Algorithm 1:
Input: An alternative a^* whose Dodgson score we wish to estimate, and a preference profile $R^N \in \mathcal{L}^N$.
Output: An approximation of the Dodgson score of a^*.
The algorithm:

1. Let A' be the alternatives that are not beaten by a^* in a pairwise election under R^N.
2. While $A' \neq \emptyset$:
   - Choose some $a \in A'$ arbitrarily.
   - Perform the minimal number of exchanges needed to make a^* beat a in a pairwise election.
   - Recalculate A'.
3. Return the number of exchanges performed.

Notice that step 2 in the while loop can be carried out efficiently. Indeed, it is sufficient to simply choose the $\mathrm{def}(a)$ agents that require the smallest number of exchanges in order to place a^* above a, and perform these exchanges.

Proposition 3.2.1. *Algorithm 1 is an m-approximation algorithm for the Dodgson score.*

Proof. Consider the given preference profile R^N; let $a \in A$ be the alternative that requires the maximum number t of exchanges in order to have a^* beat a in a pairwise election. The Dodgson score of a^* is at least t. On the other hand, each iteration of the algorithm's while loop clearly performs at most t exchanges, and there are at most m iterations. □

Unfortunately, it is also easily seen that there are examples on which Algorithm 1 gives an $\Omega(m)$ approximation. We now turn our attention to a second simple combinatorial algorithm. The input and output of the algorithm are the same as before.

Algorithm 2:

1. Let A' be the alternatives that are not beaten by a^* in a pairwise election under R^N.
2. While $A' \neq \emptyset$:
   - Move a^* upward by one position in the preferences of all the agents (unless a^* is already ranked highest).
   - Recalculate A'.
3. Return the number of exchanges performed.

Proposition 3.2.2. *Algorithm 2 is an n-approximation algorithm for the Dodgson score.*

Proof. Consider the minimal sequence of exchanges that makes a^* a Condorcet winner, and denote the length of this sequence (which is, in fact, a^*'s Dodgson score) by t. For every $i \in N$, denote by s_i^* the position of a^* as a result of this sequence in the preferences of agent i (where m is the top ranking position, and 1 is the lowest ranking). Let s_i be the position of a^* in agent i's ranking after t iterations of the algorithm's while loop. It is self evident that for all $i \in N$, $s_i \geq s_i^*$. Therefore, after at most t iterations a^* certainly becomes a Condorcet winner, and the algorithm halts. We conclude that the number of exchanges the algorithm makes is at most $t \cdot n$. □

Algorithm 2's worst-case approximation ratio is also $\Omega(n)$. Indeed, it is easy to find an example where a^* needs only one exchange to become a Condorcet winner, but a single iteration of the algorithm leads to $\Omega(n)$ exchanges.

3.2.1 The Randomized Rounding algorithm

Bartholdi et al. [9] provide an integer linear programming (ILP) formulation for the Dodgson score. The number of constraints and variables in their program depends solely on the number of alternatives. Therefore, if the number of alternatives is constant, the program is solvable in polynomial time using the algorithm of Lenstra [89]. However, if the number of alternatives is not constant, the LP is of gargantuan size.[1]

Fortunately, it is easy to modify the abovementioned ILP to obtain a program of polynomial size. As before, let $a^* \in A$ be the alternative whose score we wish to compute. Let the variables of the program be $x_j^i \in \{0,1\}$ for all $i \in N$ and $j \in \{0,\ldots,m-1\}$; $x_j^i = 1$ if and only if a^* is pushed by j positions in the ranking of agent i. Define constants $e_{ja}^i \in \{0,1\}$, for all $i \in N$, $j \in \{0,\ldots,m-1\}$,

[1]Note that there is also an efficient solution if the number of agents n is constant; indeed, brute force search requires checking $\mathcal{O}(m^n)$ possibilities.

and $a \in A \setminus \{a^*\}$, which depend on the given preference profile; $e^i_{ja} = 1$ iff pushing a^* by j positions in the ranking of agent i makes a^* gain an *additional* vote against a (note that $e^i_{ja} = 0$ for all j if $a^* R^i a$). Once again, let $\mathrm{def}(a)$ be the deficit of a^* with respect to a, i.e., the number of agents a^* must gain in order to defeat a in a pairwise election. The ILP that computes the Dodgson score of a^* is given by:

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j} j \cdot x^i_j \\ \text{subject to} \quad & \forall i \in N,\ \sum_j x^i_j = 1 \\ & \forall a \in A \setminus \{a^*\},\ \sum_{i,j} x^i_j e^i_{ja} \geq \mathrm{def}(a) \\ & \forall i \in N, \forall j \in \{0, \ldots, m-1\},\ x^i_j \in \{0,1\} \end{aligned} \tag{3.1}$$

This ILP can be relaxed by requiring merely that $0 \leq x^i_j \leq 1$ for all i and j. The resulting linear program (LP) can be solved efficiently [78].

We are now ready to present our randomized rounding algorithm. Its input and output are as before.

Randomized Rounding Algorithm

1. Solve the relaxed LP given by (3.1) to obtain a solution $\mathbf{x}$.
2. For $k = 1, \ldots, \alpha \cdot \log m$ (where $\alpha > 0$ is a constant to be chosen later)
   - For all $i \in N$, randomly and independently (from other agents and other iterations) choose a value X^i_k, such that $X^i_k = j$ with probability x^i_j.
3. For all $i \in N$, set $X^i_{max} = \max_k X^i_k$.
4. Let $\mathcal{X}'$ be the solution that moves a^* upwards in the ranking of i by X^i_{max} positions; return $\mathrm{cost}(\mathcal{X}') = \sum_{i \in N} X^i_{max}$.

We remark that if a^* is a Condorcet winner from the outset, clearly the algorithm will calculate a score of 0 (with probability 1). Therefore, if we defined a new (randomized) SCF, which elects the alternative with minimal score according to the algorithm, this SCF would satisfy the Condorcet criterion.

Theorem 3.2.3. *For any input a^* and R^N with m alternatives, the randomized rounding algorithm returns a $4\alpha \cdot \log m$-approximation of the Dodgson score of a^* with probability at least 1/2.*

The proof of the theorem is quite similar to the analysis of the randomized rounding algorithm for Set Cover [145, pp. 120-122], with one prominent additional argument, namely the application of Lemma 3.2.4.

Proof of Theorem 3.2.3. Fix some iteration k of the algorithm's for loop. Let $X^i = X^i_k$, $i \in N$, be independent discrete random variables such that $X^i = j$ with probability x^i_j. Consider the sequence

of exchanges induced by the variables X^i, i.e., each agent $i \in N$ moves a^* upward by j places with probability x^i_j. As a result of the constraint $\forall i \in N,\ \sum_j x^i_j = 1$, these are legal random variables. Moreover, let $\mathcal{X}$ be the chosen sequence of exchanges, and denote the optimal fractional solution of the LP by $\mathrm{OPT}_f = \sum_{i,j} j \cdot x^i_j$; it holds that

$$\mathbb{E}[\mathrm{cost}(\mathcal{X})] = \mathbb{E}\left[\sum_{i \in N} X^i\right] = \mathrm{OPT}_f \quad . \tag{3.2}$$

Now, fix some alternative $a \neq a^*$. We wish to bound the probability that a^* does not beat a after the exchanges given by $\mathcal{X}$ are made in R^N.

Let Y^i, $i \in N$, be independent Bernoulli trials, such that $Y^i = 1$ iff aR^ia^*, and a^* is moved above a in the preferences of agent i. In other words, $Y^i = 1$ if agent i becomes an additional agent that ranks a^* above a as a result of the exchanges. We want to provide an upper bound on $\Pr[\sum_{i \in N} Y^i < \mathrm{def}(a)]$. Denote

$$p^i = \sum_{j:\ e^i_{ja}=1} x^i_j \quad .$$

Notice that $Y^i = 1$ with probability p^i, so $\mathbb{E}[\sum_i Y^i] = \sum_i p^i$. Moreover, by the constraint $\forall a \in A \setminus \{a^*\},\ \sum_{i,j} x^i_j e^i_{ja} \geq \mathrm{def}(a)$, we have that $\sum_i p^i \geq \mathrm{def}(a)$. We now employ a deceivingly intuitive but nontrivial result:

Lemma 3.2.4 (Jogdeo and Samuels [73])**.** *Let $Y^1, \ldots, Y^n$ be independent heterogeneous Bernoulli trials. Suppose that $\mathbb{E}[\sum_i Y^i]$ is an integer. Then*

$$\Pr\left[\sum_i Y^i < \mathbb{E}\left[\sum_i Y^i\right]\right] < 1/2 \quad .$$

Since $\mathrm{def}(a)$ is an integer, and $\mathbb{E}[\sum_i Y^i] = \sum_i p^i \geq \mathrm{def}(a)$, it follows from the lemma that:

$$\Pr[a \text{ not beaten in } \mathcal{X}] = \Pr\left[\sum_i Y^i < \mathrm{def}(a)\right] < 1/2 \quad .$$

At this point, we choose the value of the constant α to be such that $2^{\alpha \log m} \geq 4m$. Note that if $m \geq 4$, we can choose $\alpha \leq 2$. As in the algorithm, set $X^i_{max} = \max_k X^i_k$. Denote by $\mathcal{X}'$ the induced sequence of exchanges. It holds that a is not beaten in a pairwise election under $\mathcal{X}'$ only if a is not beaten under the exchanges obtained in each one of the $\alpha \cdot \log m$ individual iterations. Therefore,

$$\Pr[a \text{ not beaten in } \mathcal{X}'] < \left(\frac{1}{2}\right)^{\alpha \cdot \log m} \leq \frac{1}{4m} \quad .$$

By the union bound we get:[2]

$$\Pr[a^* \text{ is not a Condorcet winner in } \mathcal{X}'] \leq m \cdot \frac{1}{4m} = 1/4 \quad . \tag{3.3}$$

[2]Strictly speaking, we can use $m - 1$ instead of m.

$X_1^i, \ldots, X_{\alpha \log m}^i$ are i.i.d. random variables; it holds that

$$X_{max}^i = \max_k X_k^i \leq \sum_k X_k^i \quad ,$$

and thus

$$\mathbb{E}\left[X_{max}^i\right] \leq \mathbb{E}\left[\sum_k X_k^i\right] = \alpha \cdot \log m \cdot \mathbb{E}[X_1^i] \quad . \tag{3.4}$$

Therefore, by the linearity of expectation,

$$\begin{aligned}
\mathbb{E}[\text{cost}(\mathcal{X}')] &= \mathbb{E}\left[\sum_i X_{max}^i\right] \\
&\leq \alpha \cdot \log m \cdot \mathbb{E}\left[\sum_i X_1^i\right] \\
&= \alpha \cdot \log m \cdot \mathbb{E}[\text{cost}(\mathcal{X})] \\
&= \alpha \cdot \log m \cdot \text{OPT}_f \\
&\leq \alpha \cdot \log m \cdot \text{OPT} \quad ,
\end{aligned}$$

where OPT is the Dodgson score of a^*, i.e., the optimal integral solution to the ILP (3.1).

By Markov's inequality we have that

$$\Pr[\text{cost}(\mathcal{X}') > \text{OPT} \cdot 4\alpha \cdot \log m] \leq 1/4 \quad . \tag{3.5}$$

We now apply the union bound once again on (3.3) and (3.5), and obtain that with probability at least $1/2$, a^* is a Condorcet winner under $\mathcal{X}'$ and, at the same time, $\text{cost}(\mathcal{X}') \leq \text{OPT} \cdot 4 \cdot \alpha \cdot \log m$. This completes the proof of Theorem 3.2.3. □

Note that it is possible to verify in polynomial time whether the output of the algorithm is, at the same time, a valid solution (i.e., a^* is a Condorcet winner) and a $4\alpha \cdot \log m$-approximation (by comparing with OPT_f). Therefore, it is possible to repeat the algorithm from scratch to improve the probability of success. The expected number of repetitions is at most 2.

3.2.2 A Matching Lower Bound

McCabe-Dansted [92] gives a polynomial-time reduction from the Minimum Dominating Set problem to the Dodgson score problem with the following property: given a graph G with k vertices, the reduction creates a preference profile with $n = \Theta(k)$ agents and $m = \Theta(k^4)$ alternatives, such that the size of the minimum dominating set of G is $\lfloor k^{-2} \text{sc}_D(a^*) \rfloor$, where $\text{sc}_D(a^*)$ is the Dodgson score of a distinguished alternative $a^* \in A$. Since the Minimum Dominating Set problem is known to be $\mathcal{NP}$-hard to approximate to within logarithmic factors [129], it follows that the Dodgson score problem is also hard to approximate to a factor of $\Omega(\log m)$. Due to the relation of Minimum Dominating Set to Minimum Set Cover, using an inapproximability result due to Feige [53], the explicit inapproximability bound can become $\left(\frac{1}{4} - \epsilon\right) \ln m$ under the assumption that problems in $\mathcal{NP}$ do not have quasi-polynomial-time algorithms.[3] This means that our randomized rounding algorithm is asymptotically optimal.

[3]Both inapproximability bounds have not been explicitly observed by McCabe-Dansted.

3.2.3 Monotonicity

We have noted that conceptually our approximation algorithm can be used an SCF in it own right. Therefore, as a short aside, we shall investigate whether it satisfies some of the properties that are considered desirable for an SCF.

Let us consider the *monotonicity* property, one of the major desiderata on the basis of which SCFs are compared. Many different notions of monotonicity can be found in the literature; for our purposes, a (score-based) SCF is *monotonic* if and only if pushing an alternative in the preferences of the agents cannot worsen the score of the alternative, that is, increase it when a lower score is desirable (as in Dodgson), or decrease it when a higher score is desirable. All prominent score-based SCFs (scoring functions, Copeland, Maximin) are monotonic; it is straightforward to see that the Dodgson and Young rules are monotonic as well.

We claim that our randomized rounding algorithm, or, more accurately, a slight variant thereof, is monotonic. Indeed, consider the variant of the algorithm where $\mathcal{X}'$ is the solution that moves a^* upward in the ranking of i by $\sum_k X_k^i$ positions rather than $\max_k X_k^i$; the cost of this solution is

$$\text{cost}(\mathcal{X}') = \sum_k \sum_{i \in N} X_k^i \quad .$$

It is easy to verify (see (3.4)) that the exact same worst-case approximation bound holds for this variant as well (although in practice its approximation ratio would usually be significantly worse).

Now, consider a situation where a^* is moved upwards in the preferences of the agents. It is obvious that this decreases the value of OPT_f. In addition, for every k, we have $\mathbb{E}\left[\sum_i X_k^i\right] = \text{OPT}_f$. Therefore, by the linearity of expectation, the expected cost of the solution produced by the algorithm $\mathbb{E}\left[\sum_k \sum_{i \in N} X_k^i\right]$ decreases as well.

3.3 Approximability of Young

Recall that the Young score of a given alternative $a^* \in A$ is the size of the largest subset of agents for which a^* is a Condorcet winner.

It is straightforward to obtain a simple ILP for the Young score problem. As before, let $a^* \in A$ be the alternative whose Young score we wish to compute. Let the variables of the program be $x^i \in \{0, 1\}$ for all $i \in N$; $x^i = 1$ iff agent i is included in the subset of agents for a^*. Define constants $e_a^i \in \{-1, 1\}$ for all $i \in N$ and $a \in A \setminus \{a^*\}$, which depend on the given preference profile; $e_a^i = 1$ iff agent i ranks a^* higher than a. The ILP that computes the Young score of a^* is given by:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in N} x^i \\ \text{subject to} \quad & \forall a \in A \setminus \{a^*\}, \ \sum_{i \in N} x^i e_a^i \geq 1 \\ & \forall i \in N, \ x^i \in \{0, 1\} \end{aligned} \tag{3.6}$$

The ILP (3.6) for the Young score is seemingly simpler than the one for the Dodgson score, given as (3.1). This might seem to indicate that the problem can be easily approximated by similar techniques. Therefore, the following result is quite surprising.

Theorem 3.3.1. *It is $\mathcal{NP}$-hard to approximate the Young score by any factor.*

This result becomes more self-evident when we notice that the Young score has the rare property of being nonmonotonic as an optimization problem, in the following sense: given a subset of agents that make a^* a Condorcet winner, it is not necessarily the case that a smaller subset of the agents would satisfy the same property. This stands in contrast to many approximable optimization problems, in which a solution which is worse than a valid solution is also a valid solution. Consider the Set Cover problem, for instance: if one adds more subsets to a valid cover, one obtains a valid cover. The same goes for the Dodgson score problem: if a sequence of exchanges makes a^* a Condorcet winner, introducing more exchanges on top of the existing ones would not undo this fact.

In order to prove the inapproximability of the Young score, we define the following problem.

NonEmptySubset
Instance: An alternative a^*, and a preference profile $R^N \in L^N$.
Question: Is there a nonempty subset of agents $C \subseteq N$, $C \neq \emptyset$, for which a^* is a Condorcet winner?

To prove Theorem 3.3.1, it is sufficient to prove that NonEmptySubset is $\mathcal{NP}$-hard. Indeed, this implies that it is $\mathcal{NP}$-hard to distinguish whether the Young score of a given alternative is zero or greater than zero, which directly entails that the score cannot be approximated.

Lemma 3.3.2. NonEmptySubset *is $\mathcal{NP}$-complete.*

Proof. The problem is clearly in $\mathcal{NP}$; a witness is given by a nonempty set of agents for which a^* is a Condorcet winner.

In order to show $\mathcal{NP}$-hardness, we present a polynomial-time reduction from the $\mathcal{NP}$-hard Exact Cover by 3-Sets (X3C) problem [58] to our problem. An instance of the X3C problem includes a finite set of elements U, $|U| = n$ (where n is divisible by 3), and a collection $\mathcal{S}$ of 3-element subsets of U, $\mathcal{S} = \{S_1, \ldots, S_k\}$, such that for every $1 \leq i \leq k$, $S_i \subseteq U$ and $|S_i| = 3$. The question is whether the collection $\mathcal{S}$ contains an *exact cover* for U, i.e., a subcollection $\mathcal{S}^* \subseteq \mathcal{S}$ of size $n/3$ such that every element of U occurs in exactly one subset in $\mathcal{S}$.

We next give the details of the reduction from X3C to NonEmptySubset. Given an instance of X3C, defined by the set U and a collection of 3-element sets $\mathcal{S}$, we construct the following instance of NonEmptySubset.

Define the set of alternatives as $A = U \cup \{a\} \cup \{a^*\}$. Let the set of agents be $N = N' \cup N''$, where N' and N'' are defined as follows. The set N' is composed of k agents, corresponding to the k subsets in $\mathcal{S}$, such that for all $i \in N'$, agent i prefers the alternatives in $U \setminus S_i$ to a^*, and prefers a^* to all the alternatives in $S_i \cup \{a\}$ (i.e., $U \setminus S_i \; R^i \; a^* R^i \; S_i \cup \{a\}$).

Subset N'' is composed of $\frac{n}{3} - 1$ agents who prefer a to a^* and a^* to U (i.e., for all $i \in N''$, $a \; R^i \; a^* \; R^i \; U$).

We next show that there is an exact cover in the given instance iff there is nonempty subset of agents for which a^* is a Condorcet winner in the constructed instance.

Sufficiency: Let $\mathcal{S}^*$ be an exact cover by 3-sets of U, and let $N^* \subseteq N'$ be the subset of agents corresponding to the $\frac{n}{3}$ subsets $S_i \in \mathcal{S}^*$. We show that a^* is a Condorcet winner for $C = N^* \cup N''$. Since $\mathcal{S}^*$ is an exact cover, for all $b \in U$ there exists exactly one agent in N^* that prefers a^* to b and $\frac{n}{3} - 1$ agents in N^* that prefer b to a^*. In addition, all $\frac{n}{3} - 1$ agents in N'' prefer a^* to b. Therefore, a^* beats b in a pairwise election.

It remains to show that a^* beats a in a pairwise election. This is true since all $\frac{n}{3}$ agents in N^* prefer a^* to a, and there are only $\frac{n}{3}-1$ agents in N'' who prefer a to a^*. It follows that a^* is a Condorcet winner for $N^* \cup N''$.

Necessity: Assume the given instance of X3C has no exact cover. We have to show that there is no subset of agents for which a^* is a Condorcet winner. Let $C \subseteq N$, $C \neq \emptyset$, and let $N^* = C \cap N'$. We distinguish between three cases.

Case 1: $|N^*| = 0$. It must hold that $C \cap N'' \neq \emptyset$. In this case, a^* loses to a in a pairwise election, since all the agents in N'' prefer a to a^*.

Case 2: $0 < |N^*| \leq \frac{n}{3}$. Since there is no exact cover, the corresponding sets S_i cannot cover U. Thus there exists $b \in U$ that is ranked higher than a^* by all agents in N^*. In order for a^* to beat b in a pairwise election, C must include at least $|N^*|+1$ agents from N''. However, this means that a beats a^* in a pairwise election (since a is ranked lower than a^* by $|N^*|$ agents, and higher than a^* by at least $|N^*|+1$ agents). It follows that a^* is not a Condorcet winner for C.

Case 3: $|N^*| > \frac{n}{3}$. Let us award each alternative $b \in A \setminus \{a^*\}$ a point for each agent that ranks it above a^*, and subtract a point for each agent that ranks it below a^*. a^* is a Condorcet winner iff the score of every other alternative, counted this way, is negative. This implies that a^* is a Condorcet winner only if for every subset $B \subseteq A$ of alternatives, the total score of the alternatives in B is at most $-|B|$.

We shall calculate the total score of the alternatives in U from the agents in N^*. Every agent in N^* prefers a^* to 3 alternatives in U and prefers $n-3$ alternatives in U to a^*. Thus, every agent in N^* contributes $(n-3)-3 = n-6$ points to the total score of U. Summing over all the agents in N^*, we have that the total score of U from N^* is $|N^*|(n-6)$. By $|N^*| > \frac{n}{3}$, we have that

$$|N^*|(n-6) \geq \left((\frac{n}{3}-1)+2\right)(n-6) = (\frac{n}{3}-1)n-6 \quad .$$

Recall that every agent in N'' prefers a^* to all alternatives in U. However, since $|N''| = \frac{n}{3}-1$, agents from N'' can only subtract $(\frac{n}{3}-1)n$ from the total score of U. We conclude that the total score of U is at least -6. Since we can assume that $|U| = n > 6$,[4] a^* cannot beat all the alternatives in U in pairwise elections. This concludes the proof. □

Theorem 3.3.1 states that the Young score cannot be efficiently approximated to any factor. The proof shows that, in fact, it is impossible to efficiently distinguish between a zero and a nonzero score. However, the proof actually shows more: it constructs a family of instances, where it is hard to distinguish between a score of zero and almost $2m/3$. Now, if one looks at an alternative formulation of the Young score problem where all the scores are scaled by an additive constant, it is no longer true that it is hard to approximate the score to *any* factor; however, the proof still shows that it is hard to approximate the Young score, even under this alternative formulation, to a factor of $\Omega(m)$.

3.4 Related Work

The agenda of approximating SCFs was recently pursued by Ailon et al. [1], Coppersmith et al. [36], and Kenyon-Mathieu and Schudy [77]. These works deal, directly or indirectly, with the Kemeny

[4] X3C is obviously tractable for a constant n, as one can examine all the families $\mathcal{S}' \subseteq \mathcal{S}$ of constant size in polynomial time.

SWF, which chooses a ranking of the alternatives instead of a single winning alternative. The Kemeny rule picks the ranking that has the maximum number of agreements with the agents' individual rankings regarding the correct order of pairs of alternatives. Ailon et al. improve the trivial 2-approximation algorithm to an involved randomized algorithm that gives an 11/7-approximation; Kenyon-Mathieu and Schudy further improve the approximation, and obtain a PTAS. Coppersmith et al. show that the Borda ranking is a 5-approximation of the Kemeny ranking. Interestingly, Klamler [80] discusses the relation between the Kemeny rule and an extension of Dodgson's rule. However, Klamler shows that the alternative ranked first by Kemeny can appear anywhere in the Dodgson ranking. This implies that approximation algorithms for Kemeny cannot be leveraged to approximate Dodgson.

Two recent works have directly put forward algorithms for the Dodgson winner problem [70, 93]. Both papers independently build upon the same basic idea: if the number of agents is significantly larger than the number of alternatives, and one looks at a uniform distribution over the preferences of the agents, with high probability one obtains an instance on which it is trivial to compute the Dodgson score of a given alternative. This directly gives rise to an algorithm with the property that Homan and Hemaspaandra [70] call *frequently self-knowingly correct*: the algorithm knows when it is definitely correct, and the algorithm is able to give a definite answer with high probability (under the assumption on the number of agents and alternatives). However, this is not an approximation algorithm in the usual sense, since the algorithm *a priori* gives up on certain instances, whereas an approximation algorithm is judged by its worst-case guarantees. In addition, this algorithm would be useless if the number of alternatives is not small compared to the number of agents.[5]

Betzler et al. [13] have investigated the parameterized computational complexity of the Dodgson and Young rules. The authors have devised a fixed parameter algorithm for exact computation of the Dodgson score, where the fixed parameter is the "edit distance", i.e., the number of exchanges. Specifically, if k is an upper bound on the Dodgson score of a given alternative, n is the number of agents, and m the number of alternatives, the algorithm runs in time $\mathcal{O}(2^k \cdot nk + nm)$. Notice that in general it may hold that $k = \Omega(nm)$. In contrast, computing the Young score is $W[2]$-complete; this implies that there is no algorithm that computes the Young score exactly, and whose running time is polynomial in n, m and only exponential in k, where the parameter k is the number of remaining votes. These results complement ours nicely, as we have also demonstrated that computing the Dodgson score is in a sense easier than computing the Young score, albeit in the context of approximation.

More distantly related to our work is research that is concerned with exactly resolving hard-to-compute SCFs by heuristic methods. Typical examples include works regarding the Kemeny rule [34] and the Slater rule [26].

Last but certainly not least, very recent subsequent work by Caragiannis et al. [21] has brought an almost complete understanding of the approximability of the Dodgson and Young rules. They have presented a deterministic algorithm that gives an $\mathcal{O}(\log m)$ approximation ratio for the Dodgson score. They have also shown that the Dodgson ranking is extremely hard to approximate. Specifically, they have shown that it is $\mathcal{NP}$-hard to distinguish whether a given alternative is the Dodgson winner or in the last $m - \mathcal{O}(\sqrt{m})$ last positions in the ranking. Finally, Caragiannis et al. have given a similar result for the Young ranking: it is hard to distinguish whether an alternative

[5]This would normally not happen in political elections, but can certainly be the case in many other settings. For instance, consider a group of agents trying to reach an agreement on a joint plan, when multiple alternative plans are available.

is in the first $\mathcal{O}(\sqrt{m})$ positions, or is ranked last.

3.5 Discussion

The work presented here and its subsequent extension [21] give rise to a promising agenda, that of studying the desirability of approximation algorithms as SCFs. Indeed, the deterministic approximation algorithm for Dodgson presented in [21] is computationally superior in every way to the one presented here: it is combinatorial rather than LP-based, and deterministic rather than randomized. However, we have argued that approximation algorithms serve as new SCFs. Therefore, it is necessary to compare the two algorithms in terms of their social choice properties.

In the algorithmic mechanism design literature, the goal is usually to design approximation algorithms that are strategyproof, namely agents cannot benefit by lying. However, the Gibbard-Satterthwaite Theorem [60, 135] precludes strategyproof SCFs. Therefore, other desiderata are looked for in SCFs.

Interestingly, it turned out that a variation on our randomized rounding algorithm is monotonic (see Section 3.2.3), whereas the deterministic algorithm is not monotonic [21]. Hence, the randomized rounding algorithm may be superior in terms of its social choice properties.

Still, there are other prominent social choice properties that are often considered, such as *homogeneity* (duplicating the electorate does not change the outcome). In addition, a stronger notion of monotonicity is often considered in the literature: pushing a winning alternative cannot change the outcome of the election. Dodgson itself is not monotonic in this sense. Is it possible to design an algorithm that approximates the Dodgson score and is monotonic in the stronger sense? We elaborate on this point in Chapter 9.

Chapter 4

Approximating Maximum Degree in a Tournament by Binary Trees

4.1 Introduction

In this chapter we again tackle the problem of choosing the "best" alternatives, this time from a *tournament*, i.e., a complete and asymmetric (dominance) relation over a set of alternatives (see Section 2.3). Such a relation for example arises from pairwise majority voting with an odd number of voters and linear preferences, and hence tournaments are intimately connected to Voting Theory and Social Choice Theory in general. In graph theoretic terms, a tournament is an orientation of a complete undirected graph, with a directed edge from a dominating alternative to a dominated one. In the presence of cycles the concept of maximality is not well-defined, and so-called tournament solutions have been devised to take over the role of singling out good alternatives. A prominent such solution, known as the Copeland solution, selects the alternatives with *maximum (out-)degree*, i.e., those that beat the largest number of other alternatives in a direct comparison. Notice that this is a reinterpretation of the Copeland SCF as defined in Chapter 2.

An interesting question concerns the implementation of a solution concept using a specific procedure. We shall specifically be interested in the well-known class of procedures given by *voting trees*. Recall (Section 2.3) that a voting tree over a set A of alternatives is a binary tree with leaves labeled by elements of A. Given a tournament T, a labeling for the internal nodes is defined recursively by labeling a node by the label of its child that beats the other child according to T (or by the unique label of its children if both have the same label). The label at the root is then deemed the winner of the voting tree given tournament T. This definition expressly allows an alternative to appear multiple times at the leaves of a tree.

A voting tree over A is said to *implement* a particular solution concept if for every tournament on A it selects an optimal alternative according to said solution concept. It has long been known that there exists no voting tree implementing the Copeland solution, i.e., one that always selects a vertex with maximum degree [102]. In this chapter, we ask a natural question from a computer science point of view: "Is there a voting tree that *approximates* the maximum degree?" More precisely, we would like to determine the largest value of α, such that for any set A of alternatives, there exists a tree Γ, which for every tournament on A selects an alternative with at least α times the maximum degree in the tournament. We will address this question both in the *deterministic* model, where Γ is a fixed voting tree, and in the *randomized* model, where voting trees are chosen

randomly according to some distribution.

4.2 The Mathematical Framework

Since in this chapter we do not have a set of agents, we denote for convenience the set of alternatives by $A = \{1, \ldots, m\}$. We refer the reader to Section 2.3 for the formal definitions associated with tournaments and voting trees, and the necessary notations.

We call a voting tree Γ *surjective* if every alternative can be elected given an appropriate tournament. Obviously, surjectivity corresponds to a very basic fairness requirement on the solution implemented by a tree. Other authors therefore view surjectivity as an inherent property of voting trees and define them accordingly (see, e.g., Moulin [102]). The sole reason we do not require surjectivity by definition is that our analysis will, on one occasion, use trees that are not necessarily surjective.

Given a tournament T and an alternative $i \in A$ we denote by $s_i = s_i(T) = |\{j \in A : iTj\}|$ the *degree* or (Copeland) *score* of i, i.e., the number of outgoing edges from this alternative, omitting T when it is clear from the context.

A voting tree Γ on A will be said to provide an approximation ratio of α (w.r.t. the maximum degree) if

$$\min_{T \in \mathcal{T}(A)} \frac{s_{\Gamma(T)}}{\max_{i \in A} s_i(T)} \geq \alpha \quad .$$

The above model can be generalized by looking at *randomizations* over voting trees according to some probability distribution. We will call a randomization *admissible* if its support contains only surjective trees. A distribution Δ over voting trees will then be said to provide a (randomized) approximation ratio of α if

$$\min_{T \in \mathcal{T}(A)} \frac{\mathbb{E}_{\Gamma \sim \Delta}[s_{\Gamma(T)}]}{\max_{i \in A} s_i(T)} \geq \alpha \quad .$$

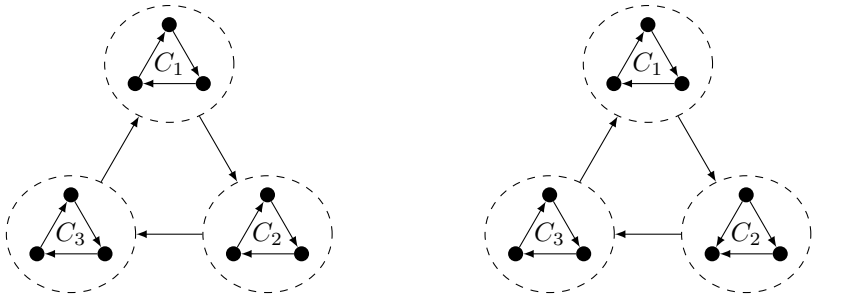
While we are of course interested in the approximation ratio achievable by admissible randomizations, it will prove useful to consider a specific class of randomizations that are not admissible, namely those that choose uniformly from the set of all voting trees with a given structure. Equivalently, such a randomization is obtained by fixing a binary tree and assigning alternatives to the leaves independently and uniformly at random, and will thus be called a *randomized voting tree*.

4.3 Upper Bounds

In this section we derive upper bounds on the approximation ratio achievable by voting trees, both in the deterministic model and in the randomized model. We build on concepts and techniques introduced by Moulin [102], and begin by quickly familiarizing the reader with these.

Given a tournament T on a set A of alternatives, we say that $C \subseteq A$ is a *component*[1] of T if for all $i_1, i_2 \in C$ and $j \in A \setminus C$, $i_1 T j$ if and only if $i_2 T j$. For a component C, denote by $\mathcal{T}_C$ the subset of tournaments that have C as a component. If $T \in \mathcal{T}_C$, we can unambiguously define a tournament T_C on $(A \setminus C) \cup \{C\}$ by replacing the component C by a single alternative. The following lemma states that for two tournaments that differ only inside a particular component, any tree chooses an alternative from that component for one of the tournaments if and only if it does for the other.

[1] Moulin [102] uses the term "adjacent set".



tournament T (C_2 regular) tournament T' (C_2 transitive)

Figure 4.1: Tournaments used in the proof of Theorem 4.3.2, illustrated for $k = 3$. A voting tree is assumed to select an alternative from C_1.

Furthermore, if an alternative outside the component is chosen for one tournament, then the same alternative has to be chosen for the other. Laslier [87] calls a solution concept satisfying these properties *weakly composition-consistent*.

Lemma 4.3.1 (Moulin [102])**.** *Let A be a set of alternatives, Γ a voting tree on A. Then, for all proper subsets $C \subsetneq A$, and for all $T, T' \in \mathcal{T}_C$,*

1. *$[T_C = T'_C]$ implies $[\Gamma(T) \in C$ if and only if $\Gamma(T') \in C]$, and*
2. *$[T_C = T'_C$ and $\Gamma(T) \in A \setminus C]$ implies $[\Gamma(T) = \Gamma(T')]$.*

We are now ready to strengthen the negative result concerning implementability of the Copeland solution [102] by showing that no deterministic tree can always choose an alternative that has a degree significantly larger than 3/4 of the maximum degree.

Theorem 4.3.2. *Let A be a set of alternatives, $|A| = m$, and let Γ be a deterministic voting tree on A with approximation ratio α. Then, $\alpha \leq 3/4 + \mathcal{O}(1/m)$.*

Proof. For ease of exposition, we assume $|A| = m = 3k + 1$ for some odd k, but the same result (up to lower order terms) holds for all values of m. Define a tournament T comprised of three components C_1, C_2, and C_3, such that for $r = 1, 2, 3$, (i) $|C_r| = k$ and the restriction of T to C_r is regular, i.e., each $i \in C_r$ dominates exactly $(k-1)/2$ of the alternatives in C_r, and (ii) for all $i \in C_r$ and $j \in C_{(r \bmod 3)+1}$, iTj. An illustration for $k = 3$ is given on the left of Figure 4.1.

Now consider any deterministic voting tree Γ on A, and assume w.l.o.g. that $\Gamma(T) \in C_1$. Define T' to be a tournament on A such that the restrictions of T and T' to $B \subseteq A$ are identical if $|B \cap C_2| \leq 1$, and the restriction of T' to C_2 is transitive; in particular, there is $i \in C_2$ such that for any $i \neq j \in C_2$, $iT'j$. An illustration for $k = 3$ is given on the right of Figure 4.1. By Lemma 4.3.1, $\Gamma(T') = \Gamma(T)$. Furthermore, T' satisfies

$$s_{\Gamma(T')} = k + \frac{(k-1)}{2} = \frac{3k}{2} - \frac{1}{2} \qquad \text{and} \qquad \max_{i \in A} s_i = 2k - 1 \quad ,$$

and thus

$$\frac{s_{\Gamma(T')}}{\max_{i \in A} s_i(T')} = \frac{3k-1}{4k-2} \leq \frac{3(k-1)+2}{4(k-1)} = \frac{3}{4} + \frac{1}{2(k-1)} \quad .$$

□

We now turn to the randomized model. It turns out that one cannot obtain an approximation ratio arbitrarily close to 1 by randomizing over large trees. We derive an upper bound for the approximation ratio by using similar arguments as in the deterministic case above, and combining them with the minimax principle of Yao [153].

Theorem 4.3.3. *Let A be a set of alternatives, $|A| = m$, and let Δ be a probability distribution over voting trees on A with an approximation ratio of α. Then, $\alpha \leq 5/6 + \mathcal{O}(1/m)$.*

The proof of this theorem is given in Appendix A.1. We point out that the theorem holds in particular for inadmissible randomizations.

4.4 A Randomized Lower Bound

A weak deterministic lower bound of $\Theta((\log m)/m)$ can be obtained straightforwardly from a balanced tree where every label appears exactly once. While balanced trees will be discussed in more detail in Section 4.5, they become increasingly unwieldy with growing height, and an improvement of this lower bound or of the deterministic upper bound given in the previous section currently seems to be out of our reach. In the remainder of the chapter, we therefore concentrate on the randomized model.

In this section we put forward our main result, a lower bound of $1/2$, up to lower order terms, for admissible randomizations over voting trees. Let us state the result formally.

Theorem 4.4.1. *Let A be a set of alternatives. Then there exists an admissible randomization over voting trees on A of size polynomial in $|A|$ with an approximation ratio of $1/2 - \mathcal{O}(1/m)$.*

In addition to satisfying the basic admissibility requirement, the randomization also has the desirable property of relying only on trees of polynomial size. This clearly facilitates its use as a computational procedure. To prove Theorem 4.4.1, we make use of a specific binary tree structure known as caterpillar trees.

4.4.1 Randomized Voting Caterpillars

We begin by inductively defining a family of binary trees that we refer to as *k-caterpillars*. The 1-caterpillar consists of a single leaf. A k-caterpillar is a binary tree, where one subtree of the root is a $(k-1)$-caterpillar, and the other subtree is a leaf. Then, a *voting k-caterpillar* on A is a k-caterpillar whose leaves are labeled by elements of A.

It is straightforward to see that an upper and lower bound of $1/2$ holds for the randomized 1-caterpillar, i.e., the uniform distribution over the m possible voting 1-caterpillars. Indeed, such a tree is equivalent to selecting an alternative uniformly at random. Since we have $\sum_{i \in A} s_i = \binom{m}{2}$, the expected score of a random alternative is $(m-1)/2$, whereas the maximum possible score is $m-1$. This randomization, however, like other randomizations over small trees that conceivably provide a good approximation ratio, is not admissible and actually puts probability one on trees that are not surjective. This leads to absurdities from a social choice point of view; for instance, in a tournament where there are both a *Condorcet winner*, an alternative that beats every other, and a *Condorcet loser*, which loses to every other alternative, the probabilities (under the above inadmissible randomization) of electing the former and the latter are equal, namely $1/m$. In contrast, any admissible randomization would elect a Condorcet winner with probability 1 given

a tournament where one exists, and would elect a Condorcet loser with probability 0 given a tournament where one exists.

To prove Theorem 4.4.1, we instead use the uniform randomization over surjective k-caterpillars, henceforth denoted k-RSC, which is clearly admissible. Theorem 4.4.1 can then be restated as a more explicit—and slightly stronger—result about the k-RSC.

Lemma 4.4.2. *Let A be a set of alternatives, $T \in \mathcal{T}(A)$. For $k \in \mathbb{N}$, denote by $p_i^{(k)}$ the probability that alternative $i \in A$ is selected from T by the k-RSC. Then, for every $\epsilon > 0$ there exists $k = k(m, \epsilon)$ polynomial in m and $1/\epsilon$ such that*

$$\sum_{i \in A} p_i^{(k)} s_i \geq \frac{m-1}{2} - \epsilon.$$

The lemma directly implies Theorem 4.4.1 by letting $\epsilon = 1$ and recalling that the maximum score is $m - 1$. The remainder of this section is devoted to the proof of this lemma. For the sake of analysis, we will use the randomized k-caterpillar, or k-RC, as a proxy to the k-RSC. We recall that the k-RC is equivalent to a k-caterpillar with labels for the leaves chosen independently and uniformly at random. In other words, it corresponds to the uniform distribution over all possible voting k-caterpillars, rather than just the surjective ones.

Clearly the k-RC corresponds to a randomization that is not admissible. In contrast to very small trees, however, like the one consisting only of a single leaf, it is straightforward to show that the distribution over alternatives selected by the RC is very close to that of the RSC.

Lemma 4.4.3. *Let $k \geq m$, and denote by $\bar{p}_i^{(k)}$ and $p_i^{(k)}$, respectively, the probability that alternative $i \in A$ is selected by the k-RC and by the k-RSC for some tournament $T \in \mathcal{T}(A)$. Then, for all $i \in A$,*

$$|\bar{p}_i^{(k)} - p_i^{(k)}| \leq \frac{m}{e^{k/m}} \quad .$$

Proof. For all $i \in A$, $|\bar{p}_i^{(k)} - p_i^{(k)}|$ is at most the probability that the k-RC does not choose a surjective tree. By the union bound, we can bound this probability by

$$\sum_{i \in A} \Pr[i \text{ does not appear in the } k\text{-RC}] \leq m \cdot \left(1 - \frac{1}{m}\right)^k \leq \frac{m}{e^{k/m}} \quad . \quad \square$$

$\square$

With Lemma 4.4.3 at hand, we can temporarily restrict our attention to the k-RC. A direct analysis of the k-RC, and in particular of the competition between the winner of the $(k-1)$-RC and a random alternative, shows that for every k, the k-RC provides an approximation ratio of at least $1/3$. It seems, however, that this analysis cannot be extended to obtain an approximation ratio of $1/2$. In order to reach a ratio of $1/2$, we shall therefore proceed by employing a second abstraction. Given a tournament T, we define a Markov chain $\mathfrak{M} = \mathfrak{M}(T)$ as follows:[2] The state

[2]Curiously, this chain bears resemblance to one previously used to define a solution concept called the Markov set (see, e.g., Laslier [87]). However, only limited attention has been given to a formal analysis of this chain, concerning properties which are different from the ones we are interested in.

space Ω of $\mathfrak{M}$ is A, and its initial distribution $\pi^{(0)}$ is the uniform distribution over Ω. The transition matrix $P = P(T)$ is given by

$$P(i,j) = \begin{cases} \frac{s_i+1}{m} & \text{if } i = j \\ \frac{1}{m} & \text{if } jTi \\ 0 & \text{if } iTj \end{cases} .$$

We claim that the distribution $\pi^{(k)}$ of $\mathfrak{M}$ after k steps is exactly the probability distribution $\bar{p}^{(k+1)}$ over alternatives selected by the $(k+1)$-RC. In order to see this, note that the 1-RC chooses an alternative uniformly at random. Then, the winner of the k-RC is the winner of the $(k-1)$-RC if the latter dominates, or is identical to, the alternative assigned to the other child of the root. This happens with probability $(s_i+1)/m$ when i is the winner of the k-RC. Otherwise the winner is some other alternative that dominates the winner of the k-RC, and each such alternative is assigned to the other child of the root with probability $1/m$.

We shall be interested in the performance guarantees given by the stationary distribution π of $\mathfrak{M}$. We first show that $\mathfrak{M}$ is guaranteed to converge to a unique such distribution, despite the fact that it is not necessarily irreducible.

Lemma 4.4.4. *Let T be a tournament. Then $\mathfrak{M}(T)$ converges to a unique stationary distribution.*

Proof (sketch). Let A be a set of alternatives. We first observe that any tournament $T \in \mathcal{T}(A)$ has a unique strongly connected component $tc(T) \subseteq A$, the *top cycle* of T, such that there is a directed path in T from every $i \in tc(T)$ to every $j \in A$. Clearly, a is a recurrent state of $\mathfrak{M} = \mathfrak{M}(T)$ if and only if $a \in tc(T)$. It follows that for every $\epsilon > 0$ there exists $k \in \mathbb{N}$ such that $\sum_{i \in tc(T)} \pi_i^{(k)} \geq 1 - \epsilon$. Since the restriction of T to $tc(T)$ is strongly connected, and since there is a positive probability of going from any state of $\mathfrak{M}$ to the same state in one step, the restriction of $\mathfrak{M}$ to $tc(T)$ is ergodic and thus has a unique stationary distribution. Moreover, $\mathfrak{M}$ is guaranteed to converge to this distribution as soon as it has reached a state in $tc(T)$, which in turn happens with probability tending to one as the number of steps tends to infinity. Finally, it is easily verified that the distribution which assigns probability zero to every $i \notin tc(T)$ and equals the stationary distribution of the restriction of $\mathfrak{M}$ to $tc(T)$ for every $i \in tc(T)$ is a stationary distribution of $\mathfrak{M}$. □

We are now ready to show that an alternative drawn from the stationary distribution will have an expected degree of at least half the maximum possible degree.

Lemma 4.4.5. *Let $T \in \mathcal{T}(A)$ be a tournament, π the stationary distribution of $\mathfrak{M}(T)$. Then*

$$\sum_{i \in A} \pi_i s_i \geq \frac{m-1}{2}.$$

To analyze π, we require the following lemma.

Lemma 4.4.6. *Let T be a tournament, π the stationary distribution of $\mathfrak{M}(T)$. Then*

$$\sum_{i=1}^{m} (2m - 2s_i - 1)\pi_i^2 = 1.$$

Proof. Let

$$q_i = 2\pi_i \cdot \left(\sum_{j:iTj} \pi_j \right) + \pi_i^2.$$

Then

$$\sum_{i=1}^{m} q_i = \sum_{i \neq j} \pi_i \pi_j + \sum_{i=1}^{m} \pi_i^2 = \left(\sum_{i=1}^{m} \pi_i \right)^2 = 1.$$

On the other hand, since π is a stationary distribution,

$$\pi_i = \frac{s_i + 1}{m} \pi_i + \frac{1}{m} \sum_{j:iTj} \pi_j,$$

and thus

$$\sum_{j:iTj} \pi_j = (m - s_i - 1) \cdot \pi_i.$$

Hence, $q_i = (2m - 2s_i - 1)\pi_i^2$, which completes the proof. □

We are now ready to prove Lemma 4.4.5.

Proof of Lemma 4.4.5. For any $i \in A$, define $w_i = m - s_i - 1$. It then holds that

$$\sum_i \pi_i s_i + \sum_i \pi_i w_i = (m-1) \sum_i \pi_i = m - 1. \tag{4.1}$$

By the Cauchy-Schwarz inequality,

$$\sum_i (2w_i + 1)\pi_i \leq \sqrt{\sum_i (2w_i + 1)} \cdot \sqrt{\sum_i (2w_i + 1)\pi_i^2}.$$

Using Lemma 4.4.6, $\sum_i (2w_i + 1)\pi_i^2 = 1$. Furthermore,

$$\sum_i (2w_i + 1) = 2m^2 - 2\binom{m}{2} - m = m^2,$$

and thus,

$$\sum_i (2w_i + 1)\pi_i \leq \sqrt{m^2} \cdot \sqrt{1} = m$$

and

$$\sum_i w_i \pi_i \leq \frac{m}{2} - \frac{\sum_i \pi_i}{2} = \frac{m-1}{2}. \tag{4.2}$$

By combining (4.1) and (4.2) we obtain

$$\sum_i \pi_i s_i \geq \frac{m-1}{2}.$$

□

The last ingredient in the proof of Lemma 4.4.2 and Theorem 4.4.1 is to show that for some k polynomial in m, the distribution over alternatives selected by the k-RC, which we recall to be equal to the distribution of $\mathfrak{M}$ after $k-1$ steps, is close to the stationary distribution of $\mathfrak{M}$. In other words, we want to show that for every tournament T, $\mathfrak{M}(T)$ is rapidly mixing.[3]

Lemma 4.4.7. *Let T be a tournament. Then, for every $\epsilon > 0$ there exists $k = k(m, \epsilon)$ polynomial in m and $1/\epsilon$, such that for all $k' > k$ and all $i \in A$, $|\pi_i^{(k')} - \pi_i| \leq \epsilon$, where $\pi^{(k)}$ is the distribution of $\mathfrak{M}(T)$ after k steps and π is the stationary distribution of $\mathfrak{M}(T)$.*

The proof of Lemma 4.4.7 works by reversibilizing the transition matrix of $\mathfrak{M}$ and then bounding the spectral gap of the reversibilized matrix via its conductance.

Proof of Lemma 4.4.7. We make use of the fact that for every tournament $T \in \mathcal{T}(A)$ and every alternative $i \in A$ with maximum degree, there exists a path of length at most two from i to any other alternative. To see this, assume for contradiction that $i \in A$ has maximum degree, and that $j \in A$ is not reachable from i in two steps. Then jTi, and for all $j' \in A$, iTj' implies jTj'. Thus, $s_j > s_i$, a contradiction. This observation implies that at any given time, $\mathfrak{M}$ either is in a state corresponding to an alternative with maximum degree, or it will reach such a state within two steps with probability at least $1/m^2$. It further implies that any alternative with maximum degree is in $tc(A)$, defined as in the proof of Lemma 4.4.4. We recall that once $\mathfrak{M}$ reaches the top cycle, it stays there indefinitely. Hence, for every $\epsilon > 0$ there exists k polynomial in m and $1/\epsilon$, such that for all $k' > k$ and all $i \notin tc(T)$, $|\pi_i^{(k')} - \pi_i| = |\pi_i^{(k')}| \leq \epsilon$, where the equality follows from the fact that the support of π is contained in $tc(T)$ (see the proof of Lemma 4.4.4).

We further observe that π is positive on $tc(T)$, i.e., for all $i \in tc(T)$, $\pi_i > 0$. Too see this, consider the largest subset of $tc(T)$ that is assigned probability zero by π, and assume that this set is nonempty. Then, for π to be a stationary distribution, no alternative in this subset can dominate an alternative in $tc(T)$ but outside the subset, contradicting the fact that $tc(T)$ is strongly connected. By all the above, we can thus focus on the restriction of $\mathfrak{M}$ to $tc(T)$. For notational convenience, we henceforth assume w.l.o.g. that $\mathfrak{M}$, rather than its restriction, is irreducible and has a stationary distribution that is positive everywhere.

Conveniently, the state space Ω of $\mathfrak{M}$ has size m, and all entries of its transition matrix P are either 0 or polynomial in m. However, there exist tournaments T such that the stationary distribution of $\mathfrak{M}(T)$ has entries that are positive but exponentially small. Furthermore, things are complicated by the fact that $\mathfrak{M}$ is usually not reversible. We follow Fill [55] in defining the *time reversal* of P as

$$\tilde{P}(i,j) = \frac{\pi_j P(j,i)}{\pi_i},$$

and the *multiplicative reversibilization* of P as $M = M(P) = P\tilde{P}$. Then, both P and $\tilde{P}$ are ergodic with stationary distribution π, and M is a reversible transition matrix that has stationary distribution π as well. Denote by $\beta_1(M)$ the second largest eigenvalue of M. Then, by Theorem 2.7 of Fill [55],

$$4\|\pi^{(k)} - \pi\|^2 \leq (\beta_1(M))^k |\Omega|, \tag{4.3}$$

[3] We might be slightly abusing terminology here, since the theory of rapidly mixing Markov chains usually considers chains with an exponential state space, which converge in time poly-logarithmic in the size of the state space. In our case the size of the state space is only m, and the mixing rate is polynomial in m.

where $\|\sigma-\pi\| = \frac{1}{2}\sum_i |\sigma_i - \pi_i|$ is the *variation distance* between a given probability mass function σ and π. Since $|\Omega| = m$, it is sufficient to show that $\beta_1(M)$ is polynomially bounded away from 1.

To this end, we will look at the *conductance*[4] of M, which measures the ability of M to leave any subset of the state space that has small weight under π. For a nonempty subset $S \subseteq A$, denote $\bar{S} = A \setminus S$ and $\pi_S = \sum_{i\in S} \pi_i$, and define $Q(i,j) = \pi_i M(i,j)$ and $Q(S,\bar{S}) = \sum_{i\in S, j\in\bar{S}} Q(i,j)$. The conductance of M is then given by

$$\Phi = \min_{S\subset A:\ \pi(S)\le 1/2} \frac{Q(S,\bar{S})}{\pi_S}.$$

It is known from the work of Sinclair and Jerrum [140] that for a Markov chain reversible with respect to a stationary distribution that is positive everywhere,

$$1 - 2\Phi \le \beta_1(A) \le 1 - \frac{\Phi^2}{2}.$$

It thus suffices to bound Φ polynomially away from 0. For any S with $\pi_S \le 1/2$ it holds that

$$\frac{Q(S,\bar{S})}{\pi_S} \ge \frac{Q(S,\bar{S})}{2\pi_S\pi_{\bar{S}}} = \frac{\sum_{i\in S, j\in\bar{S}} Q(i,j)}{2\sum_{i\in S, j\in\bar{S}} \pi_i\pi_j} \ge \min_{i\in S, j\in\bar{S}} \frac{Q(i,j)}{2\pi_i\pi_j}.$$

In our case,

$$Q(i,j) = \pi_i\left[\sum_{r\in A} P(i,r)\tilde{P}(r,j)\right] \ge \pi_i[P(i,i)\tilde{P}(i,j) + P(i,j)\tilde{P}(j,j)] \ge \frac{1}{m}[\pi_i P(i,j) + \pi_j P(j,i)]. \tag{4.4}$$

A crucial observation is that for every $i \ne j$, either $P(i,j) = 1/m$ or $P(j,i) = 1/m$, since either iTj or jTi. Now, let $i_0 \in S$ and $j_0 \in \bar{S}$ be the two alternatives for which the minimum above is attained. If $P(i_0,j_0) = 1/m$, then by (4.4),

$$\frac{Q(i_0,j_0)}{2\pi_{i_0}\pi_{j_0}} \ge \frac{\frac{\pi_{i_0}}{m^2}}{2\pi_{i_0}\pi_{j_0}} = \frac{1}{2m^2\pi_{j_0}},$$

whereas if $P(j_0,i_0) = 1/m$, then

$$\frac{Q(i_0,j_0)}{2\pi_{i_0}\pi_{j_0}} \ge \frac{1}{2m^2\pi_{i_0}}.$$

In both cases, $\Phi \ge 1/(2m^2)$, which completes the proof. □

We now have all the necessary ingredients in place.

Proof of Lemma 4.4.2 and Theorem 4.4.1. Let $\epsilon > 0$. By Lemma 4.4.3 and Lemma 4.4.7, there exists k polynomial in m and $1/\epsilon$ such that for all $i \in A$, $|p_i^{(k)} - \bar{p}_i^{(k)}| \le \epsilon/(2\binom{m}{2})$ and $|\bar{p}_i^{(k)} - \pi_i| \le \epsilon/(2\binom{m}{2})$. By the triangle inequality, $|p_i^{(k)} - \pi_i| \le \epsilon/\binom{m}{2}$. Now,

$$\sum_i \pi_i s_i - \sum_i p_i^{(k)} s_i \le \sum_i |\pi_i - p_i^{(k)}| s_i \le \frac{\epsilon}{\binom{m}{2}} \sum_i s_i = \epsilon.$$

Lemma 4.4.2 and thus Theorem 4.4.1 follow directly by Lemma 4.4.5. □

[4]The conductance is called *Cheeger constant* by Fill [55].

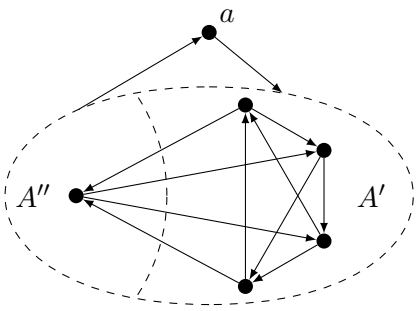


Figure 4.2: Tournament structure providing an upper bound for the randomized k-caterpillar, example for $m = 6$ and $\epsilon = 1/5$. A' and A'' contain $(1-\epsilon)(m-1)$ and $\epsilon(m-1)$ alternatives, respectively.

4.4.2 Tightness and Stability of the Caterpillar

It turns out that the analysis in the proof of Theorem 4.4.1 is tight. Indeed, since we have seen that the stationary distribution π of $\mathfrak{M}$ is very close to the distribution of alternatives chosen by the k-RSC, it is sufficient to see that π cannot guarantee an approximation ratio better than $1/2$ in expectation. Consider a set A of alternatives, and a partition of A into three sets A', A'', and $\{a\}$ such that $|A'| = (1-\epsilon)(m-1)$ and $|A''| = \epsilon(m-1)$ for some $\epsilon > 0$. Further consider a tournament $T \in \mathcal{T}(A)$ in which a dominates every alternative in A' and is itself dominated by every alternative in A'', and for which the restriction of T to $A' \cup A''$ is regular. The structure of T is illustrated in Figure 4.2.

It is easily verified that the stationary distribution π of $\mathfrak{M}(T)$ satisfies

$$\pi_a = \frac{\sum_{j:aTj} \pi_j}{m - s_a - 1} \leq \frac{1}{m - s_a - 1} \leq \frac{1}{\epsilon(m-1)},$$

and therefore,

$$\sum_i \pi_i s_i \leq \frac{1}{\epsilon(m-1)}(m-1) + \frac{\epsilon(m-1)-1}{\epsilon(m-1)} \cdot \left(\frac{m-1}{2} + 1\right) \leq \frac{m-1}{2} + \frac{1}{\epsilon} + 1.$$

Furthermore, a has degree $(1-\epsilon)(m-1)$. If we choose, say, $\epsilon = 1/\sqrt{m}$, then the approximation ratio tends to $1/2$ as m tends to infinity.

We proceed to demonstrate that the above tournament is a generic bad example. Indeed, Lemma 4.4.5 will be shown to possess the following stability property: in every tournament where π achieves an approximation ratio only slightly better than $1/2$, almost all alternatives have degree close to $m/2$, as it is the case for the example above. In particular, this implies that $\mathfrak{M}$ either provides an expected approximation ratio better than $1/2$, or selects an alternative with score around $m/2$ with very high probability.

Theorem 4.4.8. *Let $\epsilon > 0$, $m \geq 1/(2\sqrt{\epsilon})$. Let T be a tournament over a set of m alternatives, π the stationary distribution of $\mathfrak{M}(T)$. If $\sum_i \pi_i s_i = (m-1)/2 + \epsilon m$, then*

$$\left| \left\{ i \in A : \left| s_i - \frac{m}{2} \right| > \frac{3\sqrt[4]{4\epsilon}}{2} m \right\} \right| \leq \sqrt[4]{4\epsilon} \cdot m.$$

The details of the proof appear in Appendix A.2.

4.4.3 Second Order Degrees

So far we have been concerned with the Copeland solution, which selects an alternative with maximum degree. Recently, a related solution concept, sometimes referred to as *second order Copeland*, has received attention in the social choice literature(see, e.g., Bartholdi et al. [8]). Given a tournament T, this solution breaks ties with respect to the maximum degree toward alternatives i with maximum *second order degree* $\sum_{j:iTj} s_j$. Second order Copeland is the first rule, and one of only two natural voting rules, known to be computationally easy to compute but difficult to manipulate [8].

Interestingly, the same randomization studied in Section 4.4.1 also achieves a 1/2-approximation for the second order degree.

Theorem 4.4.9. *Let A be a set of alternatives, $T \in \mathcal{T}(A)$. For $k \in \mathbb{N}$, let $p_i^{(k)}$ denote the probability that alternative $i \in A$ is selected by the k-RSC for T. Then, there exists $k = k(m)$ polynomial in m such that*

$$\frac{\sum p_i^{(k)} \sum_{j:iTj} s_j}{\max_{i \in A} \sum_{j:iTj} s_j} \geq \frac{1}{2} + \Omega(1/m).$$

Clearly, the sum of degrees of alternatives dominated by an alternative i is at most $\binom{m-1}{2}$. The lower bound is then obtained from an explicit result about the second order degree of alternatives chosen by the k-RSC. Along similar lines as in the proof of Theorem 4.4.1, it suffices to prove that the stationary distribution of $\mathfrak{M}(T)$ provides an approximation. The following lemma is the second order analog of Lemma 4.4.5.

Lemma 4.4.10. *Let T be a tournament, π the stationary distribution of $\mathfrak{M}(T)$. Then,*

$$\sum_{i \in A} \left(\pi_i \sum_{j:iTj} s_j \right) \geq \frac{m^2}{4} - \frac{m}{2}.$$

It turns out that the technique used in the proof of Lemma 4.4.5, namely directly manipulating the stationary distribution equations and applying Cauchy-Schwarz, does not work for the second order degree. We instead formulate a suitable LP and bound the primal by a feasible solution to the dual. The proof of the lemma, which in turn implies Theorem 4.4.9, is given in Appendix A.3.

We further point out that the analysis is tight. Indeed, the second order degree of any alternative in a regular tournament, i.e., one where each alternative dominates exactly $(m-1)/2$ other alternatives, is $(m-1)/2 \cdot (m-1)/2 = m^2/4 - m/2 + 1/4$. Theorem 4.4.9 itself is also tight, by the example given in Section 4.4.2.

4.5 Balanced Trees

In the previous section we presented our main positive results, all of which were obtained using randomizations over caterpillars. Since caterpillars are maximally unbalanced, one would hope to do much better by looking at *balanced trees*, i.e., trees where the depth of any two leaves differs by at most one. We briefly explore this intuition. Consider a balanced binary tree where each alternative in a set A appears exactly once at a leaf. We will call such a tree a *permutation tree* on A. As we have already mentioned in the previous section, permutations trees provide a very weak deterministic lower bound. Indeed, the winning alternative must dominate the $\Theta(\log m)$

alternatives it meets on the path to the root, all of which are distinct. Since there always exists an alternative with score at least $(m-1)/2$, we obtain an approximation ratio of $\Theta((\log m)/m)$. On the other hand, no voting tree in which every two leaves have distinct labels can guarantee to choose an alternative with degree larger than the height of the tree, so the above bound is tight.

More interestingly, it can be shown that no composition of permutation trees, i.e., no tree obtained by replacing every leaf of an arbitrary binary tree by a permutation tree, can provide a lower bound better than 1/2. To see this, assume that m is a power of 2, and consider a tree Γ as above. Let Γ' be a specific permutation tree appearing as a subtree of Γ, and consider two alternatives i and j assigned to the left and right subtree of Γ', respectively. Define $C_1 \subset A$ to be the set obtained by taking all alternatives that appear in the left subtree of Γ' and replacing i by j. Similarly, let $C_2 \subset A$ be the set of all alternatives but j that appearing in the right subtree of Γ. Now define a tournament T with three components C_1, C_2, and $\{i\}$ such that iTC_1, C_1TC_2, and C_2Ti, and such that the restriction of T to C_1 is transitive. Clearly $\Gamma'(T) = i$. Furthermore, for every permutation tree Γ'' on A, $\Gamma''(T) \in C_1 \cup \{i\}$, and thus $\Gamma(T) = i$. However, $s_i = m/2$, while some element of C_1 attains the maximum degree of $m-1$. Unfortunately, larger balanced trees not built from permutation trees have so far remained elusive.

Can we obtain a better bound by randomizing? Intuitively, a randomization over large balanced trees should work well, because one would expect that the winning alternative dominate a large number of randomly chosen alternatives on the way to the root. Surprisingly, the complete opposite is the case.

In the following, we call *randomized perfect voting tree* of height k, or k-RPT, a voting tree where every leaf is at depth k and labels are assigned uniformly at random. This tree obviously corresponds to a randomization that is not admissible, but a similar result for admissible randomizations can easily be obtained by using the same arguments as before.

Theorem 4.5.1. *Let A be a set of alternatives, $|A| \geq 5$. For every $K \in \mathbb{N}$ and $\epsilon > 0$, there exists $K' \geq K$ such that the K'-RPT provides an approximation ratio of at most $\mathcal{O}(1/m)$.*

The proof of this theorem, given in Appendix A.4, constructs a tournament consisting of a 3-cycle of components and shows that the distribution over alternatives chosen by the k-RPT *oscillates* between the different components as k grows.

In Appendix A.5 we analyze higher order voting caterpillars obtained by replacing each leaf of a caterpillar of sufficiently large height by higher order caterpillars of smaller order (in particular, of order reduced by one). As in the case of the k-RPT, this construction does not provide better bounds but instead causes the approximation ratio to deteriorate.

4.6 Related Work

In economics, the problem of implementation by voting trees was introduced by Farquharson [52], and further explored, for example, by McKelvey and Niemi [95], Miller [98], Moulin [102], Herrero and Srivastava [69], Dutta and Sen [41], Srivastava and Trick [143], and Coughlan and Le Breton [37]. In particular, Moulin [102] has shown that the Copeland solution is not implementable by voting trees if there are at least 8 alternatives, while Srivastava and Trick [143] have demonstrated that it can be implemented for tournaments with up to 7 alternatives.

Laffond et al. [82] have computed the *Copeland measure* of several prominent SCCs. In contrast to the (Copeland) approximation ratio considered in this chapter, the Copeland measure is

computed with respect to the best alternative selected by the correspondence, so strictly speaking it is not a worst-case measure. More importantly, however, Laffond et al. [82] have studied properties of given correspondences, whereas we investigate the possibility of *constructing* voting trees with certain desirable properties. In this sense, our work is algorithmic in nature, while theirs is descriptive.

In theoretical computer science, the problem studied in this chapter is somewhat reminiscent of the problem of determining query complexity of graph properties (see, e.g., Rosenberg [131], Rivest and Vuillemin [130], Kahn et al. [74], King [79]). In the general model, one is given an unknown graph over a known set of vertices, and must determine whether the graph satisfies a certain property by querying the edges. The complexity of a property is then defined as the height of the smallest decision tree that checks the property. Voting trees can be interpreted as querying the edges of the tournament in parallel, and in a way that severely limits the ways in which, and the extent up to which, information can be transferred between different queries.

In the area of computational social choice, which lies at the boundary of computer science and economics, several authors have looked at the computational properties of voting trees and of various solution concepts. For example, Lang et al. [86] have characterized the computational complexity of determining different types of winners in voting trees. Procaccia et al. [126] have investigated the learnability of voting trees, as functions from tournaments to alternatives (see Chapter 5). In a slightly different context, Brandt et al. [18] have studied the computational complexity of different solution concepts, including the Copeland solution.

4.7 Discussion

Many interesting questions arise from our work. Perhaps the most enigmatic open problem in the context of this chapter concerns tighter bounds for deterministic trees. Some results for restricted classes of trees have been discussed in Section 4.5, but in general there remains a large gap between the upper bound of $3/4$ derived in Section 4.3 and the straightforward lower bound of $\Theta((\log m)/m)$.

In the randomized model our situation is somewhat better. Nevertheless, an intriguing gap remains between our upper bound of $5/6$, which holds even for inadmissible randomizations over arbitrarily large trees, and the lower bound of $1/2$ obtained from an admissible randomization over trees of polynomial size. It might be the case that the height of a k-RPT could be chosen carefully to obtain some kind of approximation guarantee. For example, one could investigate the uniform distribution over permutation trees. The analysis of this type of randomization is closely related to the theory of dynamical systems, and we expect it to be rather involved.

Part II

Elections and Computational Learning

Chapter 5

The Learnability of Social Choice Functions

5.1 Introduction

In this chapter, we consider the following setting: an entity, which we refer to as the *designer*, has in mind an SCF (which may reflect the ethics of a society). We assume that the designer is able, for each constellation of agents' preferences with which it is presented, to designate a winning alternative (perhaps with considerable computational effort). In particular, one can think of the designer's representation of the SCF as a black box that matches preference profiles to winning alternatives. This setting is relevant, for example, when a designer has in mind different properties it wants its function to satisfy; in this case, given a preference profile, the designer can specify a winning alternative that is compatible with these properties.

We would like to find a concise and easily understandable representation of the SCF the designer has in mind. We refer to this process as *automated design of SCFs*: given a specification of properties, or, indeed, of societal ethics, find an elegant SCF that implements the specification. In this chapter, we do so by learning from examples. The designer is presented with different preference profiles, drawn according to a fixed distribution. For each profile, the designer answers with the winning alternative. The number of queries presented to the designer must intuitively be as small as possible: the computations the designer has to carry out in order to handle each query might be complex, and communication might be costly.

Now, we further assume that the "target" SCF the designer has in mind, i.e., the one given as a black box, is known to belong to some family $\mathcal{F}$ of SCFs. We would like to produce a SCF from $\mathcal{F}$ that is as "close" as possible to the target function.

By "close" we mean close with respect to the fixed distribution over preference profiles. More precisely, we would like to construct an algorithm that receives pairs of the form (preferences, winner) drawn according to a fixed distribution ρ over preferences, and outputs a scoring function, such that the probability according to ρ that our scoring function and the target function agree is as high as possible. We wish, in fact, to learn scoring functions in the framework of the formal PAC (Probably Approximately Correct) learning model; a concise introduction to this model is given in Section 5.2.

In this chapter, we look at two options for the choice of $\mathcal{F}$: the family of scoring functions, and the family of voting trees (see Sections 2.2 and 2.3). These are natural choices, since both

are broad classes of functions, and both have concise representations. Choosing $\mathcal{F}$ as above, the designer could in principle translate the possibly cumbersome, unknown representation of an SCF into a succinct one that can be easily understood and computed.

Further justification for our agenda is given by noting that it might be difficult to compute an SCF on all instances, but it might be sufficient to simply calculate the election's result on typical instances. The distribution ρ can be chosen, by the designer, to concentrate on such instances.

5.2 A Crash Course on Computational Learning Theory

In this section we give a very short introduction to the PAC model and the generalized dimension of a function class. A more comprehensive (and slightly more formal) overview of the model, and results concerning the dimension, can be found in [103].

In the PAC model, the learner is attempting to learn a function $f : Z \to Y$, which belongs to a class $\mathcal{F}$ of functions from Z to Y. The learner is given a *training set*—a set $\{z_1, \ldots, z_t\}$ of points in Z, which are sampled i.i.d. (independently and identically distributed) according to a distribution ρ over the sample space Z. ρ is unknown, but is fixed throughout the learning process. In this chapter, we assume the "realizable" case, where a target function $f^*(z)$ exists, and the given training examples are in fact labeled by the target function: $\{(z_k, f^*(z_k))\}_{k=1}^t$. The *error* of a function $f \in \mathcal{F}$ is defined as

$$\mathrm{err}(f) = \Pr_{z \sim \rho}[f(z) \neq f^*(z)]. \tag{5.1}$$

$\epsilon > 0$ is a parameter given to the learner that defines the *accuracy* of the learning process: we would like to achieve $\mathrm{err}(h) \leq \epsilon$. Notice that $\mathrm{err}(f^*) = 0$. The learner is also given an *accuracy* parameter $\delta > 0$, that provides an upper bound on the probability that $\mathrm{err}(h) > \epsilon$:

$$\Pr[\mathrm{err}(h) > \epsilon] < \delta. \tag{5.2}$$

We now formalize the discussion above:

Definition 5.2.1.

1. A *learning algorithm* L is a function from the set of all training examples to $\mathcal{F}$ with the following property: given $\epsilon, \delta \in (0, 1)$ there exists an integer $s(\epsilon, \delta)$—the *sample complexity*—such that for any distribution ρ on X, if Z is a sample of size at least s where the samples are drawn i.i.d. according to ρ, then with probability at least $1 - \delta$ it holds that $\mathrm{err}(L(Z)) \leq \epsilon$.

2. L is an *efficient* learning algorithm if it always runs in time polynomial in $1/\epsilon$, $1/\delta$, and the size of the representations of the target function, of elements in X, and of elements in Y.

3. A function class $\mathcal{F}$ is *(efficiently) PAC-learnable* if there is an (efficient) learning algorithm for $\mathcal{F}$.

The sample complexity of a learning algorithm for $\mathcal{F}$ is closely related to a measure of the combinatorial richness of the class known as the generalized dimension.

Definition 5.2.2. Let $\mathcal{F}$ be a class of functions from Z to Y. We say $\mathcal{F}$ *shatters* $S \subseteq Z$ if there exist two functions $f, g \in \mathcal{F}$ such that

1. For all $z \in S$, $f(z) \neq g(z)$.

2. For all $S_1 \subseteq S$, there exists $h \in \mathcal{F}$ such that for all $z \in S_1$, $h(z) = f(z)$, and for all $z \in S \setminus S_1$, $h(z) = g(z)$.

Definition 5.2.3. Let $\mathcal{F}$ be a class of functions from a set Z to a set Y. The *generalized dimension* of $\mathcal{F}$, denoted by $D_G(\mathcal{F})$, is the greatest integer d such that there exists a set of cardinality d that is shattered by $\mathcal{F}$.

Lemma 5.2.4. *[103, Lemma 5.1] Let Z and Y be two finite sets and let $\mathcal{F}$ be a set of total functions from Z to Y. If $d = D_G(F)$, then $2^d \leq |\mathcal{F}|$.*

A function's generalized dimension provides both upper and lower bounds on the sample complexity of algorithms.

Theorem 5.2.5. *[103, Theorem 5.1] Let $\mathcal{F}$ be a class of functions from Z to Y of generalized dimension d. Let L be an algorithm such that, when given a set of t labeled examples $\{(z_k, f^*(z_k))\}_k$ of some $f^* \in \mathcal{F}$, sampled i.i.d. according to some fixed but unknown distribution over the instance space X, produces an output $f \in \mathcal{F}$ that is consistent with the training set. Then L is an (ϵ, δ)-learning algorithm for $\mathcal{F}$ provided that the sample size obeys:*

$$s \geq \frac{1}{\epsilon}\left((\sigma_1 + \sigma_2 + 3)d \ln 2 + \ln\left(\frac{1}{\delta}\right)\right) \tag{5.3}$$

where σ_1 and σ_2 are the sizes of the representation of elements in Z and Y, respectively.

Theorem 5.2.6. *[103, Theorem 5.2] Let $\mathcal{F}$ be a function class of generalized dimension $d \geq 8$. Then any (ϵ, δ)-learning algorithm for $\mathcal{F}$, where $\epsilon \leq 1/8$ and $\delta < 1/4$, must use sample size $s \geq \frac{d}{16\epsilon}$.*

5.3 Learnability of Scoring Functions

Let α be a vector of nonnegative real numbers such that $\alpha_l \geq \alpha_{l+1}$ for all $l = 1, \ldots, m-1$. Let $f_\alpha : \mathcal{L}^N \to A$ be the scoring function defined by the vector α, i.e., each agent awards α_l points to the alternative it ranks in the l'th place, and the function elects the alternative with the most points.

Since several alternatives may have maximal scores in an election, we must adopt some method of tie-breaking. Our method works as follows. Ties are broken in favor of the alternative that was ranked first by more agents; if several alternatives have maximal scores and were ranked first by the same number of agents, the tie is broken in favor of the alternative that was ranked second by more agents; and so on.[1]

Let $\mathcal{S}_m^n$ be the class of scoring functions with n agents and m alternatives. Our goal is to learn, in the PAC model, some target function $f_{\alpha^*} \in \mathcal{S}_m^n$. To this end, the learner receives a training set $\{(R_k^N, f_{\alpha^*}(R_k^N)\}_k$, where each R_k^N is drawn from a fixed distribution over $\mathcal{L}^N$; let $x_{j_k} = f_{\alpha^*}(R_k^N)$. For the profile R_k^N, we denote by $\pi_{j,l}^k$ the number of agents that ranked alternative x_j in place l. Notice that alternative x_j's score under the preference profile R_k^N is $\sum_l \pi_{j,l}^k \alpha_l$.

[1] In case several alternatives have maximal scores and identical rankings everywhere, break ties arbitrarily—say, in favor of the alternative with the smallest index.

5.3.1 Efficient Learnability of $\mathcal{S}_m^n$

Our main goal in this section is to prove the following theorem.

Theorem 5.3.1. *For all $n, m \in \mathbb{N}$, the class $\mathcal{S}_m^n$ is efficiently PAC-learnable.*

By Theorem 5.2.5, in order to prove Theorem 5.3.1 it is sufficient to validate the following two claims: 1) that there exists an algorithm which, for any training set, runs in time polynomial in n, m, and the size of the training set, and outputs a scoring function which is consistent with the training set (assuming one exists); and 2) that the generalized dimension of the class $\mathcal{S}_m^n$ is polynomial in n and m.

Remark 5.3.2. It is possible to prove Theorem 5.3.1 by using a transformation between scoring functions and sets of linear threshold functions. Indeed, it is well-known that the VC dimension (the restriction of the generalized dimension to boolean-valued functions) of linear threshold functions over $\mathcal{F}^d$ is $d+1$. In principle, it is possible to transform a scoring function into a linear threshold function that receives (generally speaking) vectors of rankings of alternatives as input. Given a training set of profiles, we could transform it into a training set of rankings and use a learning algorithm.

However, we are interested in producing an accurate scoring function according to a distribution ρ on preference profiles, which represents typical profiles. It is possible to consider a many-to-one mapping between distributions over profiles and distributions over the abovementioned vectors of rankings. Unfortunately, when this procedure is used, it is nontrivial to guarantee that the learned SCF succeeds according to the original distribution ρ. Moreover, this procedure seems to require an increase in sample complexity compared to the analysis given below. Therefore, we proceed with the more "direct" agenda outlined above and detailed below.

It is rather straightforward to construct an efficient algorithm that outputs consistent scoring functions. Given a training set, we must choose the parameters of our scoring function in a way that, for any example, the score of the designated winner is at least as large as the scores of other alternatives. Moreover, if ties between the winner and a loser would be broken in favor of the loser, then the winner's score must be strictly higher than the loser's. Our algorithm, given as Algorithm 5.3.1, simply formulates all the constraints as linear inequalities, and solves the resulting linear program. The first part of the algorithm is meant to handle tie-breaking. Recall that $x_{j_k} = f_{\alpha^*}(R_k^N)$.

A linear program can be solved in time that is polynomial in the number of variables and inequalities; it follows that Algorithm 5.3.1's running time is polynomial in n, m, and the size of the training set.

Remark 5.3.3. Notice that any vector α with a polynomial representation can be scaled to an equivalent vector of integers which is also polynomially representable. In this case, the scores are always integral. Thus, instead of using a strict inequality in the LP's first set of constraints, we can use a weak inequality with an additive term of 1.

Remark 5.3.4. Although the transformation between learning scoring functions and learning linear threshold functions mentioned in Remark 5.3.2 has some drawbacks as a learning method, results on the computational complexity of learning linear threshold functions can be leveraged to obtain computational efficiency. Indeed, well-known algorithms such as Winnow [90] suit this purpose.

Algorithm 5.3.1 Given a training set, the algorithm returns a scoring function which is consistent with the given examples, if one exists.

for $k \leftarrow 1 \dots t$ **do**
  $X_k \leftarrow \emptyset$
  for all $x_j \neq x_{j_k}$ **do** ▷ x_{j_k} is the winner in example k
    $\pi^{\Delta} \leftarrow \pi^k_{j_k} - \pi^k_j$
    $l_0 \leftarrow \min\{l : \ \pi^{\Delta}_l \neq 0\}$
    if $\pi^{\Delta}_{l_0} < 0$ **then** ▷ Ties are broken in favor of x_j
      $X_k \leftarrow X_k \cup \{x_j\}$
    end if
  end for
end for
return a feasible solution α to the following linear program:

$$\begin{array}{l} \forall k, \ \forall x_j \in X_k, \sum_l \pi^k_{j_k,l}\alpha_l \geq \sum_l \pi^k_{j,l}\alpha_l + 1 \\ \forall k, \ \forall x_j \notin X_k, \sum_l \pi^k_{j_k,l}\alpha_l \geq \sum_l \pi^k_{j,l}\alpha_l \\ \forall l = 1, \dots, m-1 \quad \alpha_l \geq \alpha_{l+1} \\ \forall l, \ \alpha_l \geq 0 \end{array}$$

Remark 5.3.5. Algorithm 5.3.1 can also be used to check, with high probability, if the SCF the designer has in mind is indeed a scoring function, as described (in a different context) by Kalai [75] (we omit the details here). This further justifies the setting in which the SCF the designer has in mind is known to be a scoring function.

So, it remains to demonstrate that the generalized dimension of $\mathcal{S}^n_m$ is polynomial in n and m. The following lemma shows this.

Lemma 5.3.6. *The generalized dimension of the class $\mathcal{S}^n_m$ is at most m:*

$$D_G(\mathcal{S}^n_m) \leq m.$$

Proof. According to Definition 5.2.3, we need to show that any set of cardinality $m+1$ cannot be shattered by $\mathcal{S}^n_m$. Let $S = \{R^N_k\}_{k=1}^{m+1}$ be such a set, and let h, g be the two social choice functions that disagree on all preference profiles in S. We shall construct a subset $S_1 \subseteq S$ such that there is no scoring function f_α that agrees with h on S_1 and agrees with g on $S \setminus S_1$.

Let us look at the first preference profile from our set, R^N_1. We shall assume without loss of generality that $h(R^N_1) = x_1$, while $g(R^N_1) = x_2$, and that in R^N_1 ties are broken in favor of x_1. Let α be some parameter vector. If we are to have $h(R^N_1) = f_\alpha(R^N_1)$, it must hold that

$$\sum_{l=1}^{m} \pi^1_{1,l} \cdot \alpha_l \geq \sum_{l=1}^{m} \pi^1_{2,l} \cdot \alpha_l, \tag{5.4}$$

whereas if we wanted f_α to agree with g we would want the opposite:

$$\sum_{l=1}^{m} \pi^1_{1,l} \cdot \alpha_l < \sum_{l=1}^{m} \pi^1_{2,l} \cdot \alpha_l \tag{5.5}$$

More generally, we define, with respect to the profile R_k^N, the vector π_Δ^k as the vector whose l'th coordinate is the difference between the number of times the winner under h and the winner under g were ranked in the l'th place:[2]

$$\pi_\Delta^k = \pi_{h(R_k)}^k - \pi_{g(R_k)}^k. \tag{5.6}$$

Now we can concisely write necessary conditions for f_α agreeing with h or g, respectively, by writing:[3]

$$\pi_\Delta^k \cdot \alpha \geq 0 \tag{5.7}$$

$$\pi_\Delta^k \cdot \alpha \leq 0 \tag{5.8}$$

Notice that each vector π_Δ^k has exactly m coordinates. Since we have $m+1$ such vectors (corresponding to the $m+1$ profiles in S), there must be a subset of vectors that is linearly dependent. We can therefore express one of the vectors as a linear combination of the others. Without loss of generality, we assume that the first profile's vector can be written as a combination of the others with parameters β_k, not all 0:

$$\pi_\Delta^1 = \sum_{k=2}^{m+1} \beta_k \cdot \pi_\Delta^k \tag{5.9}$$

Now, we shall construct our subset S_1 of preference profiles, on which f_α agrees with h, as follows:

$$S_1 = \{k \in \{2, \ldots, m+1\} : \ \beta_k \geq 0\} \tag{5.10}$$

Suppose, by way of contradiction, that f_α agrees with h on R_k^N for $k \in S_1$, and with g on the rest. We shall examine the value of $\pi_\Delta^1 \cdot \alpha$:

$$\pi_\Delta^1 \cdot \alpha = \sum_{k=2}^{m+1} \beta_k \cdot \pi_\Delta^k \cdot \alpha = \sum_{k \in S_1} \beta_k \cdot \pi_\Delta^k \cdot \alpha + \sum_{k \notin S_1 \cup \{1\}} \beta_k \cdot \pi_\Delta^k \cdot \alpha \geq 0 \tag{5.11}$$

The last inequality is due to the construction of S_1—whenever β_k is negative, the sign of $\pi_\Delta^k \cdot \alpha$ is non-positive (f_α agrees with g), and whenever β_k is positive, the sign of $\pi_\Delta^k \cdot \alpha$ is non-negative (agreement with h).

Therefore, by equation (5.5), we have that $f(R_1^N) \neq x_2 = g(R_1^N)$. However, it holds that $1 \notin S_1$, and we assumed that f_α agrees with g outside S_1—this is a contradiction. □

Theorem 5.3.1 is thus proven. The upper bound on the generalized dimension of S_m^n is quite tight: in the next subsection we show a lower bound of $m-3$.

5.3.2 Lower Bound for the Generalized Dimension of $\mathcal{S}_m^n$

Theorem 5.2.6 implies that a lower bound on the generalized dimension of a function class is directly connected to the complexity of learning it. In particular, a tight bound on the dimension gives us an almost exact idea of the number of examples required to learn a scoring function. Therefore, we wish to bound $D_G(\mathcal{S}_m^n)$ from below as well.

[2]There is some abuse of notation here; if $h(R_k^N) = x_l$ then by $\pi_{h(R_k)}^k$ we mean π_l^k.

[3]In all profiles except R_1^N, we are indifferent to the direction in which ties are broken.

Theorem 5.3.7. *For all $n \geq 4$, $m \geq 4$, $D_G(\mathcal{S}_m^n) \geq m-3$.*

Proof. We shall produce an example set of size $m-3$ which is shattered by $\mathcal{S}_m^n$. Define a preference profile R_l^N, for $l = 3, \ldots, m-1$, as follows. For all l, the agents $1, \ldots, n-1$ rank alternative x_j in place j, i.e., they vote $x_1 R_l^i x_2 R_l^i \cdots R_l^i x_m$. The preferences R_l^n (the preferences of agent n in profile R_l^N) are defined as follows: alternative x_2 is ranked in place l, alternative x_1 is ranked in place $l+1$; the other alternatives are ranked arbitrarily by agent n. For example, if $m = 5$, $n = 6$, the preference profile R_3^N is:

R_3^1	R_3^2	R_3^3	R_3^4	R_3^5	R_3^6
x_1	x_1	x_1	x_1	x_1	x_3
x_2	x_2	x_2	x_2	x_2	x_4
x_3	x_3	x_3	x_3	x_3	x_2
x_4	x_4	x_4	x_4	x_4	x_1
x_5	x_5	x_5	x_5	x_5	x_5

Lemma 5.3.8. *For any scoring function f_α with $\alpha_1 = \alpha_2 \geq 2\alpha_3$ it holds that:*

$$f_\alpha(R_l^N) = \begin{cases} x_1 & \alpha_l = \alpha_{l+1} \\ x_2 & \alpha_l > \alpha_{l+1} \end{cases}$$

Proof. We shall first verify that x_2 has maximal score. x_2's score is at least $(n-1)\alpha_2 = (n-1)\alpha_1$. Let $j \geq 3$; x_j's score is at most $(n-1)\alpha_3 + \alpha_1$. Thus, the difference is at least $(n-1)(\alpha_1 - \alpha_3) - \alpha_1$. Since $\alpha_1 = \alpha_2 \geq 2\alpha_3$, this is at least $(n-1)(\alpha_1/2) - \alpha_1 > 0$, where the last inequality holds for $n \geq 4$.

Now, under preference profile R_l^N, x_1's score is $(n-1)\alpha_1 + \alpha_{l+1}$ and x_2's score is $(n-1)\alpha_1 + \alpha_l$. If $\alpha_l = \alpha_{l+1}$, the two alternatives have identical scores, but x_1 was ranked first by more agents (in fact, by $n-1$ agents), and thus the winner is x_1. If $\alpha_l > \alpha_{l+1}$, then x_2's score is strictly higher—hence in this case x_2 is the winner. □

Armed with Lemma 5.3.8, we will now prove that the set $\{R_l^N\}_{l=3}^{m-1}$ is shattered by $\mathcal{S}_m^n$. Let α^1 be such that $\alpha_1^1 = \alpha_2^1 \geq 2\alpha_3^1 = 2\alpha_4^1 = \cdots = 2\alpha_m^1$, and α^2 be such that $\alpha_1^1 = \alpha_2^1 \geq 2\alpha_3^1 > 2\alpha_4^1 > \cdots > 2\alpha_m^1$. By the lemma, for all $l = 3, \ldots, m-1$, $f_{\alpha^1}(R_l^N) = x_1$, and $f_{\alpha^2}(R_l^N) = x_2$.

Let $T \subseteq \{3, \ldots, m-1\}$. We must show that there exists α such that $f_\alpha(R_l^N) = x_1$ for all $l \in T$, and $f_\alpha(R_l^N) = x_2$ for all $l \notin T$. Indeed, configure the parameters such that $\alpha_1 = \alpha_2 > 2\alpha_3$, and $\alpha_l = \alpha_{l+1}$ iff $l \in T$. The result follows directly from Lemma 5.3.8. □

5.4 Learnability of Voting Trees

Recall that a voting tree on A is a binary tree with leaves labeled by alternatives. To determine the winner of the election with respect to a tournament T, one must iteratively select two siblings, label their parent by the winner according to T, and remove the siblings from the tree. This process is repeated until the root is labeled, and its label is the winner of the election (see Section 2.3 for a formal definition).

In addition, recall that a preference profile R^N of a set of agents N induces a tournament $T \in \mathcal{T}(A)$ as follows: aTb (i.e., *a dominates b*) if and only if a majority of agents prefer a to b. Thus, a voting tree is in particular an SCF. However, for the purposes of this section (and similarly

to Chapter 4) it is sufficient to regard voting trees as functions $f : \mathcal{T}(A) \to A$, that is, we will disregard the set of agents and simply consider the dominance relation T on A. We shall hereinafter refer to functions $f : \mathcal{T}(A) \to A$ as *pairwise SCFs*.

Let us therefore denote the class of voting trees over m alternatives by $\mathcal{V}_m$; we emphasize the the class depends only on m. We would like to know what the sample complexity of learning functions in $\mathcal{V}_m$ is. To elaborate a bit, since we think of voting trees as functions from $\mathcal{T}$ to A, the sample space is $\mathcal{T}$.

5.4.1 Large Voting Trees

In this section, we will show that in general, the answer to the above question is that the complexity is exponential in m. We will prove this by relying on Theorem 5.2.6; the theorem implies that in order to prove such a claim, it is sufficient to demonstrate that the generalized dimension of $\mathcal{V}_m$ is at least exponential in m. This is the task we presently turn to.

Theorem 5.4.1. *$D_G(\mathcal{V}_m)$ is exponential in m.*

Proof. Without loss of generality, we let $m = 2k+2$. We will associate every distinct binary vector $v = \langle v_1, \ldots, v_k \rangle \in \{0,1\}^k$ with a distinct example in our set of tournaments $S \subseteq \mathcal{T}$. To prove the theorem, we will show that $\mathcal{V}_m$ shatters this set S of size 2^k.

Let the set of alternatives be:

$$A = \{a, b, x_1^0, x_1^1, x_2^0, x_2^1, \ldots, x_k^0, x_k^1\}.$$

For every vector $v \in \{0,1\}^k$, define a tournament T_v as follows: for $i = 1, \ldots, k$, if $v_i = 0$, we let $x_i^0 T_v b T_v x_i^1$; otherwise, if $v_i = 1$, then $x_i^1 T_v b T_v x_i^0$. In addition, for all tournaments T_v, and all $i = 1, \ldots, k$, $j = 0, 1$, a beats x_i^j, but a loses to b. We denote by S the set of these 2^k tournaments.[4] Let f be the constant function b, i.e., a voting tree which consists of only the node b; let g be the constant function a. We must prove that for every $S_1 \subseteq S$, there is a voting tree such that b wins for every tournament in S_1 (in other words, the tree agrees with f), and a wins for every tournament in $S \setminus S_1$ (the tree agrees with g). Consider the tree in Figure 5.1, which we refer to as the i'th *2-gadget*.

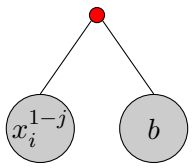


Figure 5.1: 2-gadget

With respect to this tree, b wins a tournament $T_v \in S$ iff $v_i = j$. Indeed, if $v_i = j$, the $x_i^j T_v b T_v x_i^{1-j}$, and in particular b beats x_i^{1-j}; if $v_i \neq j$, then $x_i^{1-j} T_v b T_v x_i^j$, so b loses to x_i^{1-j}.

Let $v \in \{0,1\}^k$. We will now use the 2-gadget to build a tree where b wins only the tournament $T_v \in S$, and loses every other tournament in S. Consider a balanced tree such that the deepest nodes in the tree are in fact 2-gadgets (as in Figure 5.2). As before, b wins in the i'th 2-gadget iff $v_i = j$. We will refer to this tree as a v-gadget.

[4] The relations described above are not complete, but the way they are completed is of no consequence.

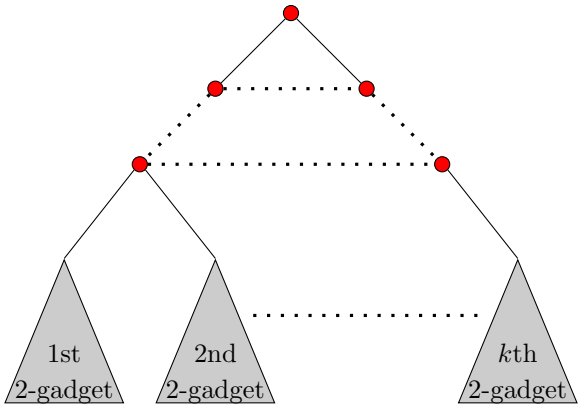


Figure 5.2: v-gadget

Now, notice that if b wins in each of the 2-gadgets (and this is the case in the tournament T_v), then b is the winner of the entire election. On the other hand, let $v' \neq v$, i.e., there exists $i \in \{1, \ldots, k\}$ such that w.l.o.g. $0 = v'_i \neq v_i = 1$. Then it holds that $x_i^0 T_{v'} b T_{v'} x_i^1$; this implies that x_i^0 wins in the i'th 2-gadget. x_i^0 proceeds to win the entire election, unless it is beaten in some stage by some other alternative x_l^j—but this must be also an alternative that beats b, as it survived the l'th 2-gadget. In any case, b cannot win the election.

Consider the small extension, in Figure 5.3, of the v-gadget, which (for lack of a better name) we call the v-gadget*.

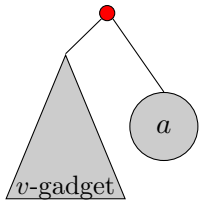


Figure 5.3: v-gadget*

Recall that, in every tournament in S, a beats any alternative x_j^i but loses to b. Therefore, by our discussion regarding the v-gadget, b wins the election described by the v-gadget* only in the tournament T^v; for any other tournament in S, alternative a wins the election.

We now present a tree and prove that it is as required, i.e., in any tournament in S_1, b is the winner, and in any tournament in $S \setminus S_1$, a prevails. Let us enumerate the tournaments in S_1:

$$S_1 = \{T_{v_1}, \ldots, T_{v_r}\}.$$

We construct a balanced tree, as in Figure 5.4, where the bottom levels consist of the v_l-gadgets*, for $l = 1, \ldots, r$.

Let $T_{v_l} \in S_1$. What is the result of this tournament in the election described by this tree? First, note that b prevails in the v_l-gadget*. The only alternatives that can reach any level above the gadgets are a and b, and b always beats a. Therefore, b proceeds to win the election. Conversely,

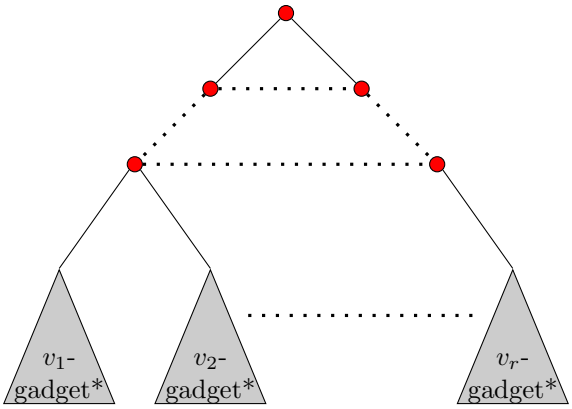


Figure 5.4: The constructed tree

let $T_v \in S \setminus S_1$. Then a survives in *every* v_l-gadget*, for $l = 1, \ldots, r$. a surely proceeds to win the entire election.

We have shown that $\mathcal{V}_m$ shatters S, thus completing the proof. □

Remark 5.4.2. Even if we restrict our attention to the class of balanced voting trees (corresponding to a playoff schedule), the dimension of the class is still exponential in m. Indeed, any unbalanced tree can be transformed to an identical (as an SCF) balanced tree. If the tree's height is h, this can be done by replacing every leaf at depth $d < h$, labeled by an alternative a, by a balanced subtree of height $d-h$ in which all the leaves are labeled by a. This implies that the class of balanced trees can shatter any sample which is shattered by $\mathcal{V}_m$.

Remark 5.4.3. The proof we have just completed, along with Lemma 5.2.4, imply that the number of different pairwise SCFs that can be represented by trees is double exponential in m, which highlights the high expressiveness of voting trees.

5.4.2 Small Voting Trees

In the previous section, we have seen that in general, a large number of examples is needed in order to learn voting trees in the PAC model. This result relied on the number of leaves in the trees being exponential in the number of alternatives. However, in many realistic settings one can expect the voting tree to be compactly represented, and in particular one can usually expect the number of leaves to be at most polynomial in m. Let us denote by $\mathcal{V}_m^{(k)}$ the class of voting trees over m alternatives, with at most k leaves. Our goal in this section is to prove the following theorem.

Theorem 5.4.4. $D_G\left(\mathcal{V}_m^{(k)}\right) = \mathcal{O}(k \log m + k \log k)$.

This theorem implies, in particular, that if the number of leaves k is polynomial in m, then the dimension of $\mathcal{V}_m^{(k)}$ is polynomial in m. In turn, this implies by Lemma 5.2.5 that the sample complexity of $\mathcal{V}_m^{(k)}$ is only polynomial in m. In other words, given a training set of size polynomial in m, $1/\epsilon$ and $1/\delta$, any algorithm that returns some tree consistent with the training set is an (ϵ, δ)-learning algorithm for $\mathcal{V}_m^{(k)}$.

To prove the theorem, we require the following straightforward lemma.

Lemma 5.4.5. *$|\mathcal{V}_m^{(k)}| \leq k \cdot m^k \cdot C_{k-1}$, where C_k is the k'th Catalan number, given by*

$$C_k = \frac{1}{k+1}\binom{2k}{k}.$$

Proof. The number of voting trees with exactly k leaves is at most the number of binary tree structures multiplied by the number of possible assignments of alternatives to leaves. The number of assignments is clearly bounded by m^k. Moreover, it is well known that the number of rooted ordered binary trees with k leaves is the $(k-1)$ Catalan number. So, the total number of voting trees with exactly k leaves is bounded by $m^k \cdot C_{k-1}$, and the number of voting trees with *at most* k leaves is at most $k \cdot m^k \cdot C_{k-1}$. □

We are now ready to prove Theorem 5.4.4.

Proof of Theorem 5.4.4. By Lemma 5.4.5, we have that

$$|\mathcal{V}_m^{(k)}| \leq k \cdot m^k \cdot C_{k-1}.$$

Therefore, by Lemma 5.2.4:

$$D_G(\mathcal{V}_m^{(k)}) \leq \log |\mathcal{V}_m^{(k)}| = \mathcal{O}(k \log m + k \log k).$$

□

5.4.3 Computational Complexity

In the previous section, we have restricted our attention to voting trees where the number of leaves is polynomial in k. We have demonstrated that the dimension of this class is polynomial in m, which implies that the sample complexity of the class is polynomial in m. Therefore, any algorithm that is consistent with a training set of polynomial size is a suitable learning algorithm (Theorem 5.2.5).

It seems that the significant bottleneck, especially in the setting of automated SCF design (finding a compact representation for a SCF that the designer has in mind), is the number of queries posed to the designer, so in this regard we are satisfied that realistic voting trees are learnable. Nonetheless, in some contexts we may also be interested in computational complexity: given a training set of polynomial size, how computationally hard is it to find a voting tree which is consistent with the training set?

In this section we explore the above question. We will assume hereinafter that the structure of the voting tree is known *a priori*. This is an assumption that we did not make before, but observe that, at least for balanced trees, Theorems 5.4.1 and 5.4.4 hold regardless. We shall try to determine how hard it is to find an assignment to the leaves which is consistent with the training set. We will refer to the computational problem as Tree-SAT (pun intended).

Definition 5.4.6. In the Tree-SAT problem, we are given a binary tree, where some of the leaves are already labeled by alternatives, and a training set that consists of pairs (T_j, x_{i_j}), where $T_j \in \mathcal{T}$ and $x_{i_j} \in A$. We are asked whether there exists an assignment of alternatives to the rest of the leaves which is consistent with the training set, i.e., for all j, the winner in T_j with respect to the tree is x_{i_j}.

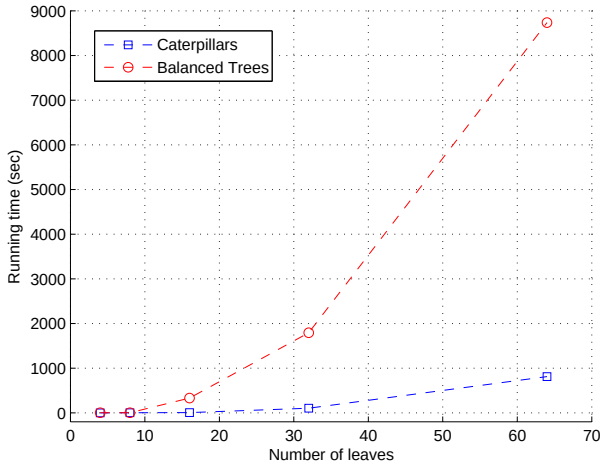


Figure 5.5: Time to find a satisfying assignment

Notice that in our formulation of the problem, some of the leaves are already labeled. However, it is reasonable to expect any efficient algorithm that finds a consistent tree, given that one exists, to be able to solve the TREE-SAT problem. Hence, an $\mathcal{NP}$-hardness result implies that such an algorithm is not likely to exist. This is actually the case.

Theorem 5.4.7. TREE-SAT *is* $\mathcal{NP}$*-complete.*

Despite Theorem 5.4.7, whose proof is delegated to Appendix B.1, it seems that in practice, solving the TREE-SAT problem is sometimes possible; we shall empirically demonstrate this.

Our simulations were carried out as follows. Given a fixed tree structure, we randomly assigned alternatives (out of a pool of 32 alternatives) to the leaves of the tree. We then used this tree to determine the winners in 20 random tournaments over our 32 alternatives. Next, we measured the time it took to find some assignment to the leaves of the tree (not necessarily the original one) which is consistent with the training set of 20 tournaments. We repeated this procedure 10 times for each number of leaves in $\{4, 8, 16, 32, 64\}$, and took the average of all ten runs.

The problem of finding a consistent tree can easily be represented as a constraint satisfaction problem, or in particular as a SAT problem. Indeed, for every node, one simply has to add one constraint per tournament which involves the node and its two children. To find a satisfying assignment, we used the SAT solver zChaff. The simulations were carried out on a PC with a Pentium D (dual core) CPU, running Linux, with 2GB of RAM and a 2.8GHz clock speed.

We experimented with two different tree structures. The first is seemingly the simplest: the caterpillar trees defined in Chapter 4. The second is intuitively the most complicated: a balanced tree. Notice that, given that the number of leaves is k, the number of nodes in both cases is $2k-1$. The simulation results are shown in Figure 5.5.

In the case of balanced trees, it is indeed hard to find a consistent tree. Adding more sample tournaments would add even more constraints and make the task harder. However, in most elections the number of alternatives is usually not above several dozen, and the problem may still be solvable. Furthermore, the problem is far easier with respect to caterpillars (even though the reduction in

Theorem 5.4.7 builds trees that are "almost caterpillars"). Therefore, we surmise that for many tree structures, it may be practically possible (in terms of the computational effort) to find a consistent assignment, even when the input is relatively large, while for others the problem is quite computationally hard even in practice.

5.5 On Learning SCFs "Close" to Target Functions

Heretofore, we have concentrated on learning SCFs that are known to be either scoring functions or voting trees. In particular, we have assumed that there is a scoring function or a voting tree that is consistent with the given training set.

In this section, we push the envelope by asking the following question: given examples that are consistent with some general SCF, is it possible to learn a scoring function or a small voting tree that is "close" to the target function?

Mathematically we are in asking whether there exist target SCFs f^* such that $\min_{f_\alpha \in \mathcal{S}_m^n} \text{err}(f_\alpha)$, or $\min_{f \in \mathcal{V}_m^{(k)}} \text{err}(f)$ (polynomial k), is large. This of course depends on the underlying distribution ρ. In the rest of this section, the implicit assumption is that ρ is the simplest nontrivial distribution over profiles, namely the uniform distribution. Nevertheless, the uniform distribution usually does not reflect real preferences of agents; this is an assumption we are making for the sake of analysis. In light of this discussion, the definition of distance between SCFs is going to be the fraction of preference profiles on which the two functions disagree.

Definition 5.5.1. an SCF $f : \mathcal{L}^N \to A$ is an *α-approximation* of an SCF g iff f and g agree on an α-fraction of the possible preference profiles:

$$|\{R^N \in \mathcal{L}^N : \ f(R^N) = g(R^N)\}| \geq \alpha \cdot (m!)^n.$$

In other words, the question is: given a training set $\{(R_k^N, f(R_j^N)\}_k$, where $f : \mathcal{L}^N \to A$ is some SCF, how hard is it to learn a scoring function or a voting tree that α-approximates f, for α that is close to 1?

It turns out that the answer is: it is impossible. We shall first give an extreme example for the case of scoring functions. Indeed, there are SCFs that disagree with any scoring function on almost all of the preference profiles; if the target function f is such a function, it is impossible to find, and of course impossible to learn, a scoring function that is "close" to f.

In order to see this, consider the following SCF that we call *flipped veto*: each agent awards one point to the alternative it ranks *last*; the winner is the alternative with the most points. This function is of course not reasonable as a preference aggregation method, but still—it is a valid SCF.

Proposition 5.5.2. *Let f_α be a scoring function. Then f_α is at most a $1/m$-approximation of flipped veto.*

Proof. Let R^N be a preference profile such that $f_\alpha(R^N) = \text{flipped veto}(R^N) = x^*$, for some $x^* \in A$. Define a set $B_{R^N} \subseteq \mathcal{L}^N$ as follows: each profile in the set is obtained by switching the place of an alternative $x \in A$, $x \neq x^*$, with the place of x^*, in the ordering of each agent that did not rank x^* last.[5] For a preference profile $R_1^N \in B_{R^N}$ that was obtained by switching x with x^*, clearly the winner under flipped veto is still x^*, as this function takes into account only alternatives ranked

[5]It cannot be the case that all agents ranked x^* last, by our tie-breaking assumption.

last. In addition, under f_α, the score of x in R_1^N is at least as large as the score of x^* in R^N (agents that have not switched the two alternatives are ones that rank x^* last, and the score of the other alternatives remains unchanged); hence $f_\alpha(R_1^N) = x$. It follows that for any preference profile in B_{R^N}, f_α and flipped veto do not agree.

We claim that for any two preference profiles R_1^N and R_2^N on which f_α and flipped veto agree, it holds that $B_{R_1^N} \cap B_{R_2^N} = \emptyset$. Indeed, assume that there exists $R^N \in B_{R_1^N} \cap B_{R_2^N}$. Assume first that the winner in both profiles is x^*. It cannot be the case that the same alternative was switched with x^* in order to obtain R^N from both R_1^N and R_2^N—that would imply R_1^N and R_2^N are identical. Therefore, assume w.l.o.g. that x_1 was switched with x^* in R_1^N (only in the rankings of agents that did not rank x^* last), and x_2 was switched with x^* in R_2^N. But this means that both x_1 and x_2 are winners in R^N under f_α (by the fact that x^* was a winner in both R_1^N and R_2^N)—a contradiction.

In addition, in any two preference profiles R_1^N and R_2^N such that

$$f_\alpha(R_1^N) = \text{flipped veto}(R_1^N) = x^*,$$

and

$$f_\alpha(R_2^N) = \text{flipped veto}(R_2^N) = x^{**},$$

it holds that $B_{R_1^N} \cap B_{R_2^N} = \emptyset$, as flipped veto elects x^* in all profiles in $B_{R_1^N}$, but elects x^{**} in all profiles in $B_{R_2^N}$.

It follows that for every preference profile on which f_α and flipped veto agree, there are at least $m-1$ distinct profiles on which the two SCFs disagree; this proves the proposition. □

We shall now formulate our main result for this Section. The theorem states that almost every SCF cannot be approximated by a factor better than $\frac{1}{2}$ by any small family of SCFs. We shall subsequently see that the theorem holds for small voting trees as well as scoring functions.

Theorem 5.5.3. *Let $\mathcal{F}_m^n$ be a family of SCFs of size exponential in n and m, and let $\epsilon, \delta > 0$. For large enough values of n and m, at least a $(1-\delta)$-fraction of the SCFs $f : \mathcal{L}^n \to \{x_1, \ldots, x_m\}$ satisfy the following property: no SCF in $\mathcal{F}_m^n$ is a $(1/2+\epsilon)$-approximation of f.*

Proof. We will surround each SCF $f \in \mathcal{F}_m^n$ with a "ball" $B(f)$, which contains all the SCFs for which f is a $(1/2+\epsilon)$-approximation. We will then show that the union of all these balls covers at most a δ-fraction of the set of the space of SCFs. This implies that for at least a $(1-\delta)$-fraction of the SCFs, no scoring function is a $(1/2+\epsilon)$-approximation.

For a given f, what is the size of $B(f)$? As there are $(m!)^n$ possible preference profiles, the ball contains functions that do not agree with f on at most $(1/2-\epsilon)(m!)^n$ preference profiles. For a profile on which there is disagreement, there are m options to set the image under the disagreeing function.[6] Therefore,

$$|B(f)| \leq \binom{(m!)^n}{(1/2-\epsilon)(m!)^n} m^{(1/2-\epsilon)(m!)^n}. \tag{5.12}$$

How large is this expression? Let $B'(f)$ be the set of all SCFs that disagree with f on *exactly*

[6] This way, we also take into account SCFs that agree with f on more than $(1/2+\epsilon)(m!)^n$ profiles.

$(1/2+\epsilon)(m!)^n$ preference profiles. It holds that

$$\begin{aligned}|B'(f)| &= \binom{(m!)^n}{(1/2+\epsilon)(m!)^n}(m-1)^{(1/2+\epsilon)(m!)^n} \\ &= \binom{(m!)^n}{(1/2-\epsilon)(m!)^n}((m-1)^{1+2\epsilon})^{1/2(m!)^n} \\ &\geq \binom{(m!)^n}{(1/2-\epsilon)(m!)^n}m^{1/2(m!)^n},\end{aligned} \tag{5.13}$$

where the last inequality holds for a large enough m. But since the total number of SCFs, $m^{(m!)^n}$, is greater than the number of functions in $B'(f)$, we have:

$$\frac{m^{(m!)^n}}{B(f)} \geq \frac{B'(\alpha)}{B(\alpha)} \geq \frac{\binom{(m!)^n}{(1/2-\epsilon)(m!)^n}m^{1/2(m!)^n}}{\binom{(m!)^n}{(1/2-\epsilon)(m!)^n}m^{(1/2-\epsilon)(m!)^n}} = m^{\epsilon(m!)^n}. \tag{5.14}$$

Therefore

$$B(f) \leq \frac{m^{(m!)^n}}{m^{\epsilon(m!)^n}} = m^{(1-\epsilon)(m!)^n}. \tag{5.15}$$

If the union of balls is to cover at least a δ-fraction of the set of SCFs, we must have $|\mathcal{F}_m^n| \cdot m^{(1-\epsilon)(m!)^n} \geq \delta \cdot m^{(m!)^n}$; equivalently, it must hold that $|\mathcal{F}_m^n| \geq \delta \cdot m^{\epsilon(m!)^n}$. However, by the assumption $|\mathcal{F}_m^n|$ is only exponential in n and m (rather than double exponential), so for large enough values of n and m, the above condition does not hold. □

Notice that the number of distinct voting trees with k leaves, as SCFs $f : \mathcal{L}^N \to A$ where $|A| = m$, is bounded from above for any number of agents n by the expression given in Lemma 5.4.5, namely $k \cdot m^k \cdot C_{k-1}$. Therefore, we have as a corollary from Theorem 5.5.3:

Corollary 5.5.4. *For large enough values of n and m, almost all SCFs cannot be approximated by $\mathcal{V}_m^{(k)}$, k polynomial in m, to a factor better than $\frac{1}{2}$.*

In order to obtain a similar corollary regarding scoring functions, we require the following lemma, which may be of independent interest.

Lemma 5.5.5. *There exists a polynomial $p(n,m)$ such that for all $n, m \in \mathbb{N}$, $|\mathcal{S}_m^n| \leq 2^{p(n,m)}$.*

Proof. It is true that there are an infinite number of ways to choose the vector α that defines a scoring function. Nevertheless, what we are really interested in is the number of *distinct* scoring functions. For instance, if $\alpha^1 = 2\alpha^2$, then $f_{\alpha^1} \equiv f_{\alpha^2}$, i.e., the two vectors define the same SCF.

It is clear that two scoring functions f_{α^1} and f_{α^2} are distinct only if the following condition holds: there exist two alternatives $x_{j_1}, x_{j_2} \in C$, and a preference profile R^N, such that $f_{\alpha^1}(R^N) = x_{j_1}$ and $f_{\alpha^2}(R^N) = x_{j_2}$. This holds only if there exist two alternatives x_{j_1} and x_{j_2} and a preference profile R^N such that under α^1, x_{j_1}'s score is strictly greater than x_{j_2}'s, and under α_2, either x_{j_2}'s score is greater or the two alternatives are tied, and the tie is broken in favor of x_{j_2}.

Now, assume R^N induces rankings π_{j_1} and π_{j_2}. The conditions above can be written as

$$\sum_l \pi_{j_1,l}\alpha_l^1 > \sum_l \pi_{j_2,l}\alpha_l^1, \tag{5.16}$$

$$\sum_l \pi_{j_1,l}\alpha_l^2 \leq \sum_l \pi_{j_2,l}\alpha_l^2, \tag{5.17}$$

where the inequality is an equality only if ties are broken in favor of x_{j_2}, i.e., if $l_0 = \min\{l : \pi_{j_1,l} \neq \pi_{j_2,l}\}$, then $\pi_{j_1,l} < \pi_{j_2,l}$.[7]

Let $\pi_\Delta = \pi_{j_1} - \pi_{j_2}$. As in the proof of Lemma 5.3.6, (5.16) and (5.17) can be concisely rewritten as

$$\pi_\Delta \cdot \alpha^1 > 0 \geq \pi_\Delta \cdot \alpha^2, \tag{5.18}$$

where the inequality is an equality only if the first nonzero position in π_Δ is negative.

In order to continue, we opt to reinterpret the above discussion geometrically. Each point in $\mathbb{R}^m$ corresponds to a possible choice of parameters α. Now, each possible choice of π_Δ is the normal to a hyperplane. These hyperplanes partition the space into cells: the vectors in the interior of each cell agree on the signs of dot products with all vectors π_Δ. More formally, if α_1 and α_2 are two points in the interior of a cell, then for any vector π_Δ, $\pi_\Delta \cdot \alpha^1 > 0 \Leftrightarrow \pi_\Delta \cdot \alpha^2 > 0$. By equation (5.18), this implies that any two scoring functions f_{α^1} and f_{α^2}, where α^1 and α^2 are in the interior of the same cell, are identical.

What about points residing in the intersection of several cells? These vectors always agree with the vectors in one of the cells, as ties are broken according to rankings induced by the preference profile, i.e., according to the parameters that define our hyperplanes. Therefore, the points in the intersection can be conceptually annexed to one of the cells.

So, we have reached the conclusion that the number of distinct scoring functions is at most the number of cells. Hence, it is enough to bound the number of cells; we claim this number is exponential in n and m. Indeed, each π_Δ is an m-vector, in which every coordinate is an integer in the set $\{-n, -n+1, \ldots, n-1, n\}$. It follows that there are at most $(2n+1)^m$ possible hyperplanes. It is known [42] that given k hyperplanes in d-dimensional space, the number of cells is at most $O(k^d)$. In our case, $k \leq (2n+1)^m$ and $d = m$, so we have obtained a bound of:

$$((2n+1)^m)^m \leq (3n)^{m^2} = \left(2^{\log 3n}\right)^{m^2} = 2^{m^2 \log 3n}. \tag{5.19}$$

□

Remark 5.5.6. This lemma implies, according to Lemma 5.2.4, that there exists a polynomial $p(n,m)$ such that for all $n, m \in \mathbb{N}$, $D_G(\mathcal{S}_m^n) \leq p(n,m)$. However, we have already obtained a tighter upper bound of m.

Finally, using Theorem 5.5.3 and Lemma 5.5.5 we obtain:

Corollary 5.5.7. *For large enough values of n and m, almost all SCFs cannot be approximated by $\mathcal{S}_m^n$ to a factor better than $\frac{1}{2}$.*

Remark 5.5.8. Proposition 5.5.2 can seemingly be circumvented by removing the requirement that in a scoring function defined by a vector α, $\alpha_l \geq \alpha_{l+1}$ for all l. Indeed, flipped veto is essentially a scoring function with $\alpha_m = 1$ and $\alpha_l = 0$ for all $l \neq m$. However, the constant SCF that always elects the same alternative has the same inapproximability ratio, even when this property of scoring functions is not taken into account. Moreover, Corollary 5.5.7 also holds when scoring functions are not assumed to satisfy this property.

[7] W.l.o.g. we disregard the case where $\pi_{j_1} = \pi_{j_2}$; the reader can verify that taking this case into account multiplies the final result by an exponential factor at most.

5.6 Related Work

Currently there exists a small body of work on learning in economic settings. Kalai [75] explores the learnability (in the PAC model) of rationalizable choice functions. These are functions which, given a set of alternatives, choose the element that is maximal with respect to some linear order. Similarly, PAC learning has been applied to computing utility functions that are rationalizations of given sequences of prices and demands [12].

Another prominent example is the paper by Lahaie and Parkes [83], which considers preference elicitation in combinatorial auctions. The authors show that preference elicitation algorithms can be constructed on the basis of existing learning algorithms. The learning model used, exact learning, differs from ours (PAC learning).

Conitzer and Sandholm [29] have studied automated mechanism design, in the more restricted setting where agents have quasi-linear preferences. They propose automatically designing a truthful mechanism for every preference aggregation setting. However, they find that, under two solution concepts, even determining whether there exists a deterministic mechanism that guarantees a certain social welfare is an $\mathcal{NP}$-complete problem. The authors also show that the problem is tractable when designing a randomized mechanism. In more recent work [31], Conitzer and Sandholm put forward an efficient algorithm for designing deterministic mechanisms, which works only in very limited scenarios. In short, our setting, goals, and methods are completely different—in the general voting context, even framing computational complexity questions is problematic, since the goal cannot be specified with reference to expected social welfare.

5.7 Discussion

It turns out (Corollaries 5.5.4 and 5.5.7) that many SCFs cannot be approximated, neither by using scoring functions nor by small voting trees. However, this negative result relied implicitly on assuming a uniform distribution over profiles. More importantly, it might be the case that some of the important families of SCFs can be approximated by scoring functions or small voting trees. Therefore, we do not rule out at this point the application of our approach to designing general SCFs by directly learning scoring functions or small voting trees that approximate them.

We mention two directions for future research. First, imagine the following scenario: the designer has in mind a huge voting tree, and would like to know whether there exists a smaller voting tree that implements the same social choice function. The same goes for scoring functions, e.g., the designer might have in mind a scoring function with huge values for components of the vector α. This is a setting closely related to ours, but our results do not hold in the alternative setting.

Second, it might prove interesting to study the learnability of larger families of SCFs that have a concise representation. One compelling example is the class of *generalized scoring functions* recently proposed by Xia and Conitzer [147].

Chapter 6

Strategyproof Regression Learning

6.1 Introduction

Following the rise of the Internet as a computational platform, machine learning problems have become increasingly dispersed, in the sense that different parts of the training set may be controlled by different computational or economic entities.

A Motivating Example. Consider an Internet search company trying to improve the performance of their search engine by learning a ranking function from examples. The ranking function is the heart of a modern search engine, and can be thought of as a mapping that assigns a real-valued score to every pair of a query and a URL. Some of the large Internet search companies currently hire Internet users, which we hereinafter refer to as "experts", to manually rank such pairs. These rankings are then pooled and used to train a ranking function. Moreover, the experts are chosen in a way such that averaging over the experts' opinions and interests presumably pleases the average Internet user.

However, different experts may have different interests and a different idea of the results a good search engine should return. For instance, take the ambiguous query "Jaguar", which has become folklore in search engine designer circles. The top answer given by most search engines for this query is the website of the luxury car manufacturer. Knowing this, an animal-loving expert may decide to give this pair a disproportionately low score, hoping to improve the relative rank of websites dedicated to the Panthera Onca. An expert who is an automobile enthusiast may counter this measure by giving automotive websites a much higher score than is appropriate. From the search company's perspective, this type of strategic manipulation introduces an undesired bias in the training set.

Setting. Our problem setting falls within the general boundaries of *statistical regression learning*. Regression learning is the task of constructing a real-valued function f based on a training set of examples, where each example consists of an input to the function and its corresponding output. In particular, the example $(\mathbf{x}, y)$ suggests that $f(\mathbf{x})$ should be equal to y. The accuracy of a function f on a given input-output pair $(\mathbf{x}, y)$ is defined using a loss function ℓ. Popular choices of the loss function are the squared loss, $\ell(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$, or the absolute loss, $\ell(f(\mathbf{x}), y) = |f(\mathbf{x}) - y|$. We typically assume that the training set is obtained by sampling i.i.d. from an underlying distribution over the product space of inputs and outputs. The overall quality of the function

constructed by the learning algorithm is defined to be its expected loss, with respect to the same distribution.

We augment this well-studied setting by introducing a set of *strategic agents*. Each agent holds as private information an individual distribution over the input space and values for the points in the support of this distribution, and measures the quality of a regression function with respect to this data. The global goal, on the other hand, is to do well with respect to the average of the individual points of view. A training set is obtained by eliciting private information from the agents, who may reveal this information untruthfully in order to favorably influence the result of the learning process.

Relation to Voting Theory. *Mechanism design* is a subfield of economics that is concerned with the question of how to incentivize agents to truthfully report their private information, also known as their type. Given potentially non-truthful reports from the agents, a mechanism determines a global solution, and possibly additional monetary transfers to and from the agents. A mechanism is said to be *strategyproof* if it is always in the agents' best interest to report their true types, and *efficient* if the solution maximizes social welfare (i.e. minimizes the overall loss). Our goal in this chapter will be to design and analyze strategyproof and efficient mechanisms for the regression learning setting.

The common assumption in the mechanism design literature is that agents have quasi-linear preferences, that is, money is available. However, this chapter mostly focuses on obtaining strategyproofness results *without payments* (see Nisan et al. [136] for an overview of results on mechanism design without money). So, the agents are essentially voting on a set of functions. We will see that it is possible to obtain strategyproofness despite the Gibbard-Satterthwaite Theorem [60, 135], since in our setting the agents cannot express all possible linear preferences over the alternatives, hence the G-S Theorem does not hold.

It should be noted that strategyproofness is essential for obtaining *any* learning theoretic bounds. Otherwise, all agents might reveal untruthful information at the same time, in a coordinated or uncoordinated way, causing the learning problem itself to be ill-defined.

6.2 The Mathematical Framework

In this section we formalize the regression learning problem described in the introduction and cast it in the framework of game theory. Some of the definitions are illustrated by relating them to the Internet search example presented in Section 6.1. Notice that the learning-theoretic model is somewhat different than the one discussed in Chapter 5, since that chapter dealt with (multi-) classification and in this chapter we deal with regression learning.

We focus on the task of learning a real-valued function over an *input space* $\mathcal{X}$. In the Internet search example, $\mathcal{X}$ would be the set of all query-URL pairs, and our task would be to learn the ranking function of a search engine. As usual, let $N = \{1, \ldots, n\}$ be a set of agents, which in our running example would be the set of all experts. For each agent $i \in N$, let o_i be a function from $\mathcal{X}$ to $\mathbb{R}$ and let ρ_i be a probability distribution over $\mathcal{X}$. Intuitively, o_i is what agent i thinks to be the correct real-valued function, while ρ_i captures the relative importance that agent i assigns to different parts of $\mathcal{X}$. In the Internet search example, o_i would be the optimal ranking function according to agent i, and ρ_i would be a distribution over query-URL pairs that assigns higher weight to queries from that agent's areas of interest.

Let $\mathcal{F}$ be a class of functions, where every $f \in \mathcal{F}$ is a function from $\mathcal{X}$ to the real line. We call $\mathcal{F}$ the *hypothesis space* of our problem, and restrict the output of the learning algorithm to functions in $\mathcal{F}$. We evaluate the accuracy of each $f \in \mathcal{F}$ using a *loss function* $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$. For a particular input-output pair $(\mathbf{x}, y)$, we interpret $\ell(f(\mathbf{x}), y)$ as the penalty associated with predicting the output value $f(\mathbf{x})$ when the true output is known to be y. As mentioned in the introduction, common choices of ℓ are the squared loss, $\ell(\alpha, \beta) = (\alpha - \beta)^2$, and the absolute loss, $\ell(\alpha, \beta) = |\alpha - \beta|$. The accuracy of a hypothesis $f \in \mathcal{F}$ is defined to be the average loss of f over the entire input space. Formally, define the *risk* associated by agent i with the function f as

$$\text{risk}_i(f) \;=\; \mathbb{E}_{\mathbf{x}\sim\rho_i}\big[\ell(f(\mathbf{x}), o_i(x))\big] \quad .$$

Clearly, this subjective definition of hypothesis accuracy allows for different agents to have significantly different valuations of different functions in $\mathcal{F}$, and it is quite possible that we will not be able to please all of the agents simultaneously. Instead, our goal is to satisfy the agents in N on average. Define J to be a random variable distributed uniformly over the elements of N. Now define the *global risk* of a function f to be the average risk with respect to all of the agents, namely

$$\text{risk}_N(f) \;=\; \mathbb{E}\left[\text{risk}_J(f)\right] \quad .$$

We are now ready to state our learning-theoretic goal formally: we would like to find a hypothesis in $\mathcal{F}$ that attains a global risk as close as possible to $\inf_{f\in\mathcal{F}} \text{risk}_N(F)$.

Even if N is small, we still have no explicit way of calculating $\text{risk}_N(f)$. Instead, we use an empirical estimate of the risk as a proxy to the risk itself. For each $i \in N$, we randomly sample m points independently from the distribution ρ_i and request their respective labels from agent i. In this way, we obtain the labeled training set $\tilde{S}_i = \{(\mathbf{x}_{i,j}, \tilde{y}_{i,j})\}_{j=1}^m$. Agent i may label the points in $\tilde{S}_i$ however it sees fit, and we therefore say that agent i *controls* (the labels of) these points. We usually denote agent i's "true" training set by $S_i = \{(\mathbf{x}_{ij}, y_{ij})\}_{j=1}^m$, where $y_{ij} = o_i(x_{ij})$. After receiving labels from all agents in N, we define the *global training set* to be the multiset $\tilde{S} = \biguplus_{i\in N} \tilde{S}_i$.

The elicited training set $\tilde{S}$ is presented to a regression learning algorithm, which in return constructs a *hypothesis* $\tilde{f} \in \mathcal{F}$. Each agent can influence $\tilde{f}$ by modifying the labels it controls. This observation brings us to the game-theoretic aspect of our setting. For all $i \in N$, agent i's private information, or type, is a vector of true labels $y_{ij} = o_i(x_{ij})$, $j = 1, \ldots, m$. The sampled points $\mathbf{x}_{ij}$, $j = 1, \ldots, m$, are exogenously given and assumed to be common knowledge. The *strategy space* of each agent then consists of all possible *values* for the labels it controls. In other words, agent i reports a labeled training set $\tilde{S}_i$. We sometimes use $\tilde{S}_{-i}$ as a shorthand for $\tilde{S} \setminus \tilde{S}_i$, the strategy profile of all agents except agent i. The space of possible outcomes is the hypothesis space $\mathcal{F}$, and the utility of agent i for an outcome $\tilde{f}$ is determined by its risk $\text{risk}_i(\tilde{f})$. More precisely, agent i chooses $\tilde{y}_{i1}, \ldots, \tilde{y}_{im}$ so as to minimize $\text{risk}_i(f)$. We follow the usual game-theoretic assumption that it does this with full knowledge of the inner workings of our regression learning algorithm, and name the resulting game the *learning game*.

Notice that under the above formalism, a regression learning algorithm is in fact a social choice function, which maps the types of the agents to a hypothesis. One of the simplest and most popular regression learning techniques is *empirical risk minimization* (ERM). The *empirical risk* associated with a hypothesis f, with respect to a sample S, is denoted by $\hat{\text{risk}}(f, S)$ and defined to be the average loss attained by f on the examples in S, i.e.

$$\hat{\text{risk}}(f, S) = \frac{1}{|S|} \sum_{(\mathbf{x},y)\in S} \ell(f(\mathbf{x}), y) \quad .$$

An ERM algorithm finds the empirical risk minimizer $\hat{f}$ within $\mathcal{F}$. More formally,

$$\hat{f} \;=\; \operatorname*{argmin}_{f \in \mathcal{F}} \; \hat{\text{risk}}(f, S) \quad .$$

A large part of this chapter will be dedicated to ERM algorithms. For some choices of loss function and hypothesis class, it may occur that the global minimizer of the empirical risk is not unique, and we must define an appropriate tie-breaking mechanism.

Since our strategy is to use $\hat{\text{risk}}(f, \tilde{S})$ as a surrogate for $\text{risk}_N(f)$, we need $\hat{\text{risk}}(f, \tilde{S})$ to be an unbiased estimator of $\text{risk}_N(f)$. A particular situation in which this can be achieved is when all agents $i \in N$ truthfully report $\tilde{y}_{ij} = o_i(\mathbf{x}_{ij})$ for all j. It is important to note that truthfulness need not come at the expense of the overall solution quality. This can be seen by a variation of the well-known revelation principle. Indeed, assume that for a given mechanism and given true inputs there is an equilibrium in which some agents report their inputs untruthfully, and which leads to an outcome that is strictly better than any outcome achievable by a strategyproof mechanism. Then we can design a new mechanism that, given the true inputs, simulates the agents' lies and yields the exact same output in equilibrium.

6.3 Degenerate Distributions

We begin our study by focusing on a special case, where each agent is only interested in a single point of the input space. Even this simple setting has interesting applications. Consider for example the problem of allocating tasks among service providers, e.g. messages to routers, jobs to remote processors, or reservations of bandwidth to Internet providers. Machine learning techniques are used to obtain a global picture of the capacities, which in turn are private information of the respective providers. Regression learning provides an appropriate model in this context, as each provider is interested in an allocation that is as close as possible to its capacity: more tasks mean more revenue, but an overload is clearly undesirable.

A concrete economic motivation for this setting is given by Perote and Perote-Peña [110]. The authors consider a monopolist trade union in some sector that has to set a common hourly wage for its members. The union collects information about the hours of work in each firm versus the firm's expected profitability, and accordingly sets a single sectorial wage per hour. The hours of work are public information, but the expected profitability is private. Workers that are more profitable might have an incentive to exaggerate their profitability in order to increase the hourly common wage.

More formally, the distribution ρ_i of agent i is now assumed to be degenerate, and the sample S_i becomes a singleton. Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the set of true input-output pairs, where now $y_i = o_i(\mathbf{x}_i)$, and $S_i = \{(\mathbf{x}_i, y_i)\}$ is the single example controlled by agent i. Each agent selects an output value $\tilde{y}_i$, and the reported (possibly untruthful) training set $\tilde{S} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^n$ is presented to a regression learning algorithm. The algorithm constructs a hypothesis $\tilde{f}$ and agent i's cost is the loss

$$\text{risk}_i(\tilde{f}) \;=\; \mathbb{E}_{\mathbf{x} \sim \rho_i}\big[\ell(\tilde{f}(\mathbf{x}), o_i(\mathbf{x}))\big] = \ell(\tilde{f}(\mathbf{x}_i), y_i)$$

on the point it controls, where ℓ is a predefined loss function. Within this setting, we examine the game-theoretic properties of ERM.

As noted above, an ERM algorithm takes as input a loss function ℓ and a training set S, and outputs the hypothesis that minimizes the empirical risk on S according to ℓ. Throughout

this section, we write $\hat{f} = \texttt{ERM}(\mathcal{F}, \ell, S)$ as shorthand for $\arg\min_{f \in \mathcal{F}} \hat{\text{risk}}(f, \ell, S)$. We restrict our discussion to loss functions of the form $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$, where $\mu : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically increasing convex function, and to the case where $\mathcal{F}$ is a convex set of functions. These assumptions enable us to cast ERM as a convex optimization problem, which are typically tractable. Most choices of ℓ and $\mathcal{F}$ that do not satisfy the above constraints may not allow for computationally efficient learning, and are therefore less interesting.

We prove two main theorems: if μ is a linear function, then ERM is group strategyproof; if on the other hand μ grows faster than any linear function, and given minimal conditions on $\mathcal{F}$, ERM is not strategyproof.

6.3.1 ERM with the Absolute Loss

In this section, we focus on the absolute loss function. Indeed, let ℓ denote the absolute loss, $\ell(a, b) = |a - b|$, and let $\mathcal{F}$ be a convex hypothesis class. Because ℓ is only weakly convex, there may be multiple hypotheses in $\mathcal{F}$ that globally minimize the empirical risk and we must add a tie-breaking step to our ERM algorithm. Concretely, consider the following two-step procedure:

1. Empirical risk minimization: calculate
$$r = \min_{f \in \mathcal{F}} \hat{\text{risk}}(f, S).$$
2. Tie-breaking: return
$$\tilde{f} = \operatorname*{argmin}_{f \in \mathcal{F} : \hat{\text{risk}}(f,S) = r} \|f\|,$$
where $\|f\|^2 = \int f^2(\mathbf{x})\, d\mathbf{x}$.

Our assumption that $\mathcal{F}$ is a convex set implies that the set of empirical risk minimizers $\{f \in \mathcal{F} : \hat{\text{risk}}(f, S) = r\}$ is also convex. The function $\|f\|$ is a strictly convex function and therefore the output of the tie-breaking step is uniquely defined.

For example, imagine that $\mathcal{X}$ is the unit ball in $\mathbb{R}^n$ and that $\mathcal{F}$ is the set of homogeneous linear functions, of the form $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$, where $\mathbf{w} \in \mathbb{R}^n$. In this case, Step 1 above can be restated as the following linear program:

$$\min_{\boldsymbol{\xi} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \xi_i \quad \text{s.t.} \quad \forall i \quad \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \leq \xi_i \quad \text{and} \quad y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle \leq \xi_i \quad .$$

The tie-breaking step can then be written as the following quadratic program with linear constraints:

$$\operatorname*{argmin}_{\boldsymbol{\xi} \in \mathbb{R}^m, \mathbf{w} \in \mathbb{R}^n} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad \textstyle\sum_{i=1}^{m} \xi_i = r \quad \text{and}$$
$$\forall i \quad \langle \mathbf{w}, \mathbf{x}_i \rangle - y_i \leq \xi_i \quad \text{and} \quad y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle \leq \xi_i \quad .$$

In our analysis, we only use the fact that $\|f\|$ is a strictly convex function of f. Any other strictly convex function can be used in its place in the tie-breaking step.

The following theorem states that ERM using the absolute loss function has excellent game-theoretic properties. More precisely, it is group strategyproof: if a member of an arbitrary coalition of agents strictly gains from a joint deviation by the coalition, then some other member must strictly

lose. It should also be noted that in our case any mechanism without payments satisfies individual rationality: if some agent does not provide values for its part of the sample, then ERM will simply return the best fit for the points of the other agents, so no agent can gain by not taking part in the mechanism.

Theorem 6.3.1. *Let N be a set of agents, $S = \uplus_{i \in N} S_i$ a training set such that $S_i = \{\mathbf{x}_i, y_i\}$ for all $i \in N$, and let ρ_i be degenerate at $\mathbf{x}_i$. Let ℓ denote the absolute loss, $\ell(a,b) = |a-b|$, and let $\mathcal{F}$ be a convex hypothesis class. Then, ERM minimizing ℓ over $\mathcal{F}$ with respect to S is group strategyproof.*

We prove this theorem below, as a corollary of the following more explicit result.

Proposition 6.3.2. *Let $\hat{S} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^m$ and $\tilde{S} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^m$ be two training sets on the same set of points, and let $\hat{f} = \texttt{ERM}(\mathcal{F}, \ell, \hat{S})$ and $\tilde{f} = \texttt{ERM}(\mathcal{F}, \ell, \tilde{S})$. If $\hat{f} \neq \tilde{f}$ then there exists $i \in N$ such that $\hat{y}_i \neq \tilde{y}_i$ and $\ell(\hat{f}(\mathbf{x}_i), \hat{y}_i) < \ell(\tilde{f}(\mathbf{x}_i), \hat{y}_i)$.*

Proof. Let U be the set of indices on which $\hat{S}$ and $\tilde{S}$ disagree, i.e. $U = \{i : \hat{y}_i \neq \tilde{y}_i\}$. We prove the claim by proving its counter-positive, i.e. we assume that $\ell(\tilde{f}(\mathbf{x}_i), \hat{y}_i) \leq \ell(\hat{f}(\mathbf{x}_i), \hat{y}_i)$ for all $i \in U$, and prove that $\hat{f} \equiv \tilde{f}$. We begin by considering functions of the form $f_\alpha(\mathbf{x}) = \alpha \tilde{f}(\mathbf{x}) + (1-\alpha)\hat{f}(\mathbf{x})$ and proving that there exists $\alpha \in (0,1]$ for which

$$\hat{\text{risk}}(\hat{f}, \tilde{S}) - \hat{\text{risk}}(\hat{f}, \hat{S}) \;=\; \hat{\text{risk}}(f_\alpha, \tilde{S}) - \hat{\text{risk}}(f_\alpha, \hat{S}) \quad . \tag{6.1}$$

For every $i \in U$, our assumption that $\ell(\tilde{f}(\mathbf{x}_i), \hat{y}_i) \leq \ell(\hat{f}(\mathbf{x}_i), \hat{y}_i)$ implies that one of the following four inequalities holds:

$$\tilde{f}(\mathbf{x}_i) \leq \hat{y}_i < \hat{f}(\mathbf{x}_i) \qquad\qquad \tilde{f}(\mathbf{x}_i) \geq \hat{y}_i > \hat{f}(\mathbf{x}_i) \tag{6.2}$$

$$\hat{y}_i \leq \tilde{f}(\mathbf{x}_i) \leq \hat{f}(\mathbf{x}_i) \qquad\qquad \hat{y}_i \geq \tilde{f}(\mathbf{x}_i) \geq \hat{f}(\mathbf{x}_i) \tag{6.3}$$

Furthermore, we assume without loss of generality that $\tilde{y}_i = \tilde{f}(\mathbf{x}_i)$ for all $i \in U$. Otherwise, we could simply change $\tilde{y}_i$ to equal $\tilde{f}(\mathbf{x}_i)$ for all $i \in U$ without changing the output of the learning algorithm. If one of the two inequalities in (6.2) holds, we set

$$\alpha_i \;=\; \frac{\hat{y}_i - \hat{f}(\mathbf{x}_i)}{\tilde{f}(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)} \quad ,$$

and note that $\alpha_i \in (0,1]$ and $f_{\alpha_i}(\mathbf{x}_i) = \hat{y}_i$. Therefore, for every $\alpha \in (0, \alpha_i]$ it holds that either

$$\tilde{y}_i \leq \hat{y}_i \leq f_\alpha(\mathbf{x}_i) < \hat{f}(\mathbf{x}_i) \quad \text{or} \quad \tilde{y}_i \geq \hat{y}_i \geq f_\alpha(\mathbf{x}_i) > \hat{f}(\mathbf{x}_i) \quad .$$

Setting $c_i = |\hat{y}_i - \tilde{y}_i|$, we conclude that for all α in $(0, \alpha_i]$,

$$\begin{aligned} &\ell(\hat{f}(\mathbf{x}_i), \tilde{y}_i) - \ell(\hat{f}(\mathbf{x}_i), \hat{y}_i) = c_i \quad \text{and} \\ &\ell(f_\alpha(\mathbf{x}_i), \tilde{y}_i) - \ell(f_\alpha(\mathbf{x}_i), \hat{y}_i) = c_i. \end{aligned} \tag{6.4}$$

Alternatively, if one of the inequalities in (6.3) holds, we have that either

$$\hat{y}_i \leq \tilde{y}_i \leq f_\alpha(\mathbf{x}_i) \leq \hat{f}(\mathbf{x}_i) \quad \text{or} \quad \hat{y}_i \geq \tilde{y}_i \geq f_\alpha(\mathbf{x}_i) \geq \hat{f}(\mathbf{x}_i) \quad .$$

Setting $\alpha_i = 1$ and $c_i = -|\tilde{y}_i - \hat{y}_i|$, we once again have that (6.4) holds for all α in $(0, \alpha_i]$. Moreover, if we choose $\alpha = \min_{i \in U} \alpha_i$, (6.4) holds simultaneously for all $i \in U$. (6.4) also holds trivially for

all $i \notin U$ with $c_i = 0$. (6.1) can now be obtained by summing both of the equalities in (6.4) over all i.

Next, we recall that $\mathcal{F}$ is a convex set and therefore $f_\alpha \in \mathcal{F}$. Since $\hat{f}$ minimizes the empirical risk with respect to $\hat{S}$ over $\mathcal{F}$, we specifically have that

$$\hat{\text{risk}}(\hat{f}, \hat{S}) \;\leq\; \hat{\text{risk}}(f_\alpha, \hat{S}) \quad . \tag{6.5}$$

Combining this inequality with (6.1) results in

$$\hat{\text{risk}}(\hat{f}, \tilde{S}) \;\leq\; \hat{\text{risk}}(f_\alpha, \tilde{S}) \quad . \tag{6.6}$$

Since the empirical risk function is convex in its first argument, we have that

$$\hat{\text{risk}}(f_\alpha, \tilde{S}) \;\leq\; \alpha\hat{\text{risk}}(\tilde{f}, \tilde{S}) \;+\; (1-\alpha)\hat{\text{risk}}(\hat{f}, \tilde{S}) \quad . \tag{6.7}$$

Replacing the left-hand side above with its lower bound in (6.6) yields $\hat{\text{risk}}(\hat{f}, \tilde{S}) \leq \hat{\text{risk}}(\tilde{f}, \tilde{S})$. On the other hand, we know that $\tilde{f}$ minimizes the empirical risk with respect to $\tilde{S}$, and specifically $\hat{\text{risk}}(\tilde{f}, \tilde{S}) \leq \hat{\text{risk}}(\hat{f}, \tilde{S})$. Overall, we have shown that

$$\hat{\text{risk}}(\hat{f}, \tilde{S}) \;=\; \hat{\text{risk}}(\tilde{f}, \tilde{S}) \;=\; \min_{f \in \mathcal{F}} \hat{\text{risk}}(f, \tilde{S}) \quad . \tag{6.8}$$

Next, we turn our attention to $\|\hat{f}\|$ and $\|\tilde{f}\|$. We start by combining (6.8) with (6.7) to get $\hat{\text{risk}}(f_\alpha, \tilde{S}) \leq \hat{\text{risk}}(\hat{f}, \tilde{S})$. Recalling (6.1), we have that $\hat{\text{risk}}(f_\alpha, \hat{S}) \leq \hat{\text{risk}}(\hat{f}, \hat{S})$. Once again using (6.5), we conclude that $\hat{\text{risk}}(f_\alpha, \hat{S}) = \hat{\text{risk}}(\hat{f}, \hat{S})$. Although $\hat{f}$ and f_α both minimize the empirical risk with respect to $\hat{S}$, we know that $\hat{f}$ was chosen as the output of the algorithm, and therefore it must hold that

$$\|\hat{f}\| \;\leq\; \|f_\alpha\| \quad . \tag{6.9}$$

Using convexity of the norm, we have $\|f_\alpha\| \leq \alpha\|\tilde{f}\| + (1-\alpha)\|\hat{f}\|$. Combining this inequality with (6.9), we get $\|\hat{f}\| \leq \|\tilde{f}\|$. On the other hand, (6.8) tells us that both $\hat{f}$ and $\tilde{f}$ minimize the empirical risk with respect to $\tilde{S}$, whereas $\tilde{f}$ is chosen as the algorithm output, so $\|\tilde{f}\| \leq \|\hat{f}\|$. Overall, we have shown that

$$\|\hat{f}\| \;=\; \|\tilde{f}\| \;=\; \min_{f \in \mathcal{F} \,:\, \hat{\text{risk}}(f, \tilde{S}) = \hat{\text{risk}}(\tilde{f}, \tilde{S})} \|f\| \quad . \tag{6.10}$$

In summary, in (6.8) we showed that both $\hat{f}$ and $\tilde{f}$ minimize the empirical risk with respect to $\tilde{S}$, and therefore both move on to the tie breaking step of the algorithm. Then, in (6.10) we showed that both functions attain the minimum norm over all empirical risk minimizers. Since the norm is strictly convex, its minimum is unique, and therefore $\hat{f} \equiv \tilde{f}$. □

To understand the intuition behind Proposition 6.3.2, as well as its relation to Theorem 6.3.1, assume that $\hat{S}$ represents the true preferences of the agents, and that $\tilde{S}$ represents the values revealed by the agents and used to train the regression function. Moreover, assume that $\hat{S} \neq \tilde{S}$. Proposition 6.3.2 states that one of two things can happen. Either $\hat{f} \equiv \tilde{f}$, i.e. revealing the values in $\tilde{S}$ instead of the true values in $\hat{S}$ does not affect the result of the learning process. In this case, the agents might as well have told the truth. Or, $\hat{f}$ and $\tilde{f}$ are different hypotheses, and Proposition 6.3.2 tells us that there must exist an agent i who lied about its true value and is strictly worse off due to his lie. Clearly, agent i has no incentive to actually participate in such a lie. This said, we can now proceed to prove the theorem.

Proof of Theorem 6.3.1. Let $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ be a training set that represents the true private information of a set N of agents and let $\tilde{S} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^m$ be the information revealed by the agents and used to train the regression function. Let $C \subseteq N$ be an arbitrary coalition of agents that have conspired to decrease some of their respective losses by lying about their values. Now define the hybrid set of values where

$$\text{for all } i \in N, \quad \hat{y}_i \;=\; \begin{cases} y_i & \text{if } i \in C \\ \tilde{y}_i & \text{otherwise} \end{cases} \quad ,$$

and let $\hat{S} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^m$. Finally, let $\hat{f} = \texttt{ERM}(\mathcal{F}, \ell, \hat{S})$ and $\tilde{f} = \texttt{ERM}(\mathcal{F}, \ell, \tilde{S})$.

If $\hat{f} \equiv \tilde{f}$ then the members of C gain nothing from being untruthful. Otherwise, Proposition 6.3.2 states that there exists an agent $i \in N$ such that $\hat{y}_i \neq \tilde{y}_i$ and $\ell(\hat{f}(\mathbf{x}_i), \hat{y}_i) < \ell(\tilde{f}(\mathbf{x}_i), \hat{y}_i)$. From $\hat{y}_i \neq \tilde{y}_i$ we conclude that this agent is a member of C. We therefore have that $\hat{y}_i = y_i$ and $\ell(\hat{f}(\mathbf{x}_i), y_i) < \ell(\tilde{f}(\mathbf{x}_i), y_i)$. This contradicts our assumption that no member of C loses from revealing $\tilde{S}$ instead of $\hat{S}$. We emphasize that the proof holds regardless of the values revealed by the agents that are not members of C, and we therefore have group strategyproofness. □

6.3.2 ERM with Other Convex Loss Functions

We have seen that performing ERM with the absolute loss is strategyproof. We now show that the same is not true for most other convex loss functions. Specifically, we examine loss functions of the form $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$, where $\mu : \mathbb{R}_+ \to \mathbb{R}$ is a monotonically increasing strictly convex function with unbounded subderivatives. Unbounded subderivatives mean that μ cannot be bounded from above by any linear function.

For example, μ can be the function $\mu(\alpha) = \alpha^d$, where d is a real number strictly greater than 1. A popular choice is $d = 2$, which induces the squared loss, $\ell(\alpha, \beta) = (\alpha - \beta)^2$. The following example demonstrates that ERM with the squared loss is not strategyproof.

Example 6.3.3. Let ℓ be the squared loss function, $\mathcal{X} = \mathbb{R}$, and $\mathcal{F}$ the class of constant function over $\mathcal{X}$. Further let $S_1 = \{(x_1, 2)\}$ and $S_2 = \{(x_2, 0)\}$. On S, ERM outputs the constant function $\hat{f}(x) \equiv 1$, and agent 1 suffers loss 1. However, if agent 1 reports its value to be 4, ERM outputs $\hat{f}(x) \equiv 2$, with loss of 0 for agent 1.

For every $\mathbf{x} \in \mathcal{X}$, let $\mathcal{F}(\mathbf{x})$ denote the *set of feasible values* at $\mathbf{x}$, formally defined as $\mathcal{F}(\mathbf{x}) = \{\, f(\mathbf{x}) \,:\, f \in \mathcal{F} \,\}$. Since $\mathcal{F}$ is a convex set, it follows that $\mathcal{F}(\mathbf{x})$ is either an interval on the real line, a ray, or the entire real line. Similarly, for a multiset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$, denote

$$\mathcal{F}(X) = \{\langle f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\rangle : f \in \mathcal{F}\} \subseteq \mathbb{R}^n \quad .$$

We then say that $\mathcal{F}$ is *full* on a multiset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ if $\mathcal{F}(X) = \mathcal{F}(\mathbf{x}_1) \times \cdots \times \mathcal{F}(\mathbf{x}_n)$. Clearly, requiring that $\mathcal{F}$ is not full on X is a necessary condition for the existence of a training set with points X where one of the agents gains by lying. Otherwise, ERM will fit any set of values for the points with an error of zero. For an example of a function class that is *not* full, consider any function class $\mathcal{F}$ on $\mathcal{X}$, $|\mathcal{F}| \geq 2$, and observe that there have to exist $f_1, f_2 \in \mathcal{F}$ and a point $\mathbf{x}_0 \in \mathcal{X}$ such that $f_1(\mathbf{x}_0) \neq f_2(\mathbf{x}_0)$. In this case, $\mathcal{F}$ is not full on any multiset X that contains two copies of $\mathbf{x}_0$.

In addition, if $|\mathcal{F}| = 1$, then any algorithm would trivially be strategyproof irrespective of the loss function. In the following theorem we therefore consider hypothesis classes $\mathcal{F}$ of size at least two which are *not* full on the set X of points of the training set.

Theorem 6.3.4. *Let $\mu : \mathbb{R}_+ \to \mathbb{R}$ be a monotonically increasing strictly convex function with unbounded subderivatives, and define the loss function $\ell(\alpha, \beta) = \mu(|\alpha - \beta|)$. Let $\mathcal{F}$ be a convex hypothesis class that contains at least two functions, and let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ be a multiset such that $\mathcal{F}$ is not full on X. Then there exist $y_1, \ldots, y_n \in \mathbb{R}$ such that, if $S = \uplus_{i \in N} S_i$ with $S_i = \{(\mathbf{x}_i, y_i)\}$, ρ_i is degenerate at $\mathbf{x}_i$, and ERM is used, there is an agent who has an incentive to lie.*

An example for a function not covered by this theorem is given by $\nu(\alpha) = \ln(1 + \mathbb{E}(\alpha))$, which is both monotonic and strictly convex, but has a derivative bounded from above by 1. We use the subderivatives of μ, rather than its derivatives, since we do not require μ to be differentiable.

As before, we actually prove a slightly stronger and more explicit claim about the behavior of the ERM algorithm. The formal proof of Theorem 6.3.4 follows as a corollary below.

Proposition 6.3.5. *Let μ and ℓ be as defined in Theorem 6.3.4 and let $\mathcal{F}$ be a convex hypothesis class. Let $\hat{S} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^m$ be a training set, where $\hat{y}_i \in \mathcal{F}(\mathbf{x}_i)$ for all i, and define $\hat{f} = \texttt{ERM}(\mathcal{F}, \ell, \hat{S})$. For each $i \in N$, one of the following conditions holds:*

1. *$\hat{f}(\mathbf{x}_i) = \hat{y}_i$.*
2. *There exists $\tilde{y}_i \in \mathbb{R}$ such that, if we define $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, \tilde{y}_i)\}$ and $\tilde{f} = \texttt{ERM}(\mathcal{F}, \ell, \tilde{S})$, $\ell(\tilde{f}(\mathbf{x}_i), \hat{y}_i) < \ell(\hat{f}(\mathbf{x}_i), \hat{y}_i)$.*

To prove the above, we first require a few technical results, which we state in the form of three lemmas. The first lemma takes the perspective of agent i and considers the case where truth-telling results in a function $\hat{f}$ such that $\hat{f}(\mathbf{x}_i) > \hat{y}_i$, i.e. agent i would like the ERM hypothesis to map $\mathbf{x}_i$ to a somewhat lower value. The second lemma then states that there exists a lie that achieves this goal. The gap between the claim of this lemma and the claim of Theorem 6.3.5 is a subtle one: merely lowering the value of the ERM hypothesis does not necessarily imply a lowering of the loss incurred by agent i. It could be the case that the lie told by agent i caused $\hat{f}(\mathbf{x}_i)$ to become too low, essentially overshooting the desired target value and increasing the loss of agent i. This point is resolved by the third lemma.

Lemma 6.3.6. *Let ℓ, $\mathcal{F}$, $\hat{S}$ and $\hat{f}$ be as defined in Theorem 6.3.5 and let $i \in N$ be such that $\hat{f}(\mathbf{x}_i) > \hat{y}_i$. Then for all $f \in \mathcal{F}$ for which $f(\mathbf{x}_i) \geq \hat{f}(\mathbf{x}_i)$, and for all $y \in \mathbb{R}$ such that $y \leq \hat{y}_i$, the dataset $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, y)\}$ satisfies $\hat{risk}(f, \tilde{S}) \geq \hat{risk}(\hat{f}, \tilde{S})$.*

Proof. Let $f \in \mathcal{F}$ be such that $f(\mathbf{x}_i) \geq \hat{f}(\mathbf{x}_i)$, let $y \in \mathbb{R}$ be such that $y \leq \hat{y}_i$, and define $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, y)\}$. We now have that

$$\begin{aligned} \hat{\text{risk}}(f, \tilde{S}) &= \hat{\text{risk}}(f, \tilde{S}_{-i}) + \ell(f(\mathbf{x}_i), \tilde{y}_i) \\ &= \hat{\text{risk}}(f, \hat{S}) - \ell(f(\mathbf{x}_i), \hat{y}_i) + \ell(f(\mathbf{x}_i), \tilde{y}_i) \\ &= \hat{\text{risk}}(f, \hat{S}) - \mu(f(\mathbf{x}_i) - \hat{y}_i) + \mu(f(\mathbf{x}_i) - \tilde{y}_i) \quad . \end{aligned} \tag{6.11}$$

Using the fact that $\hat{f}$ is the empirical risk minimizer with respect to $\hat{S}$, we can get a lower bound for the above and obtain

$$\hat{\text{risk}}(f, \tilde{S}) \;\geq\; \hat{\text{risk}}(\hat{f}, \hat{S}) - \mu(f(\mathbf{x}_i) - \hat{y}_i) + \mu(f(\mathbf{x}_i) - \tilde{y}_i) \quad .$$

The term $\hat{\text{risk}}(\hat{f}, \hat{S})$ on the right hand side can again be rewritten using (6.11), resulting in

$$\hat{\text{risk}}(f, \tilde{S}) \;\geq\; \hat{\text{risk}}(\hat{f}, \tilde{S}) + \mu(\hat{f}(\mathbf{x}_i) - \hat{y}_i) - \mu(\hat{f}(\mathbf{x}_i) - \tilde{y}_i) - \mu(f(\mathbf{x}_i) - \hat{y}_i) + \mu(f(\mathbf{x}_i) - \tilde{y}_i) \quad .$$

Denoting $a = \hat{f}(\mathbf{x}_i) - \hat{y}_i$, $b = \hat{f}(\mathbf{x}_i) - \tilde{y}_i$, $c = f(\mathbf{x}_i) - \hat{y}_i$, and $d = f(\mathbf{x}_i) - \tilde{y}_i$, we can rewrite the above as

$$\hat{\text{risk}}(f, \tilde{S}) \;\geq\; \hat{\text{risk}}(\hat{f}, \tilde{S}) + \mu(a) - \mu(b) - \mu(c) + \mu(d) \quad . \tag{6.12}$$

Noting that b, c, and d are all greater than a, and that $b + c - 2a = d - a$, we use convexity of μ to obtain

$$\begin{aligned}
\mu(a) + \mu(d) &= \left(\frac{b-a}{d-a}\mu(a) + \frac{c-a}{d-a}\mu(d)\right) + \left(\frac{c-a}{d-a}\mu(a) + \frac{b-a}{d-a}\mu(d)\right) \\
&\geq \mu\left(\frac{(b-a)a + (c-a)d}{d-a}\right) + \mu\left(\frac{(c-a)a + (b-a)d}{d-a}\right) \\
&= \mu\left(\frac{(b+c-2a)a + (c-a)(d-a)}{d-a}\right) + \\
&\qquad \mu\left(\frac{(c+b-2a)a + (b-a)(d-a)}{d-a}\right) \\
&= \mu(c) \;+\; \mu(b) \quad .
\end{aligned}$$

Combining this inequality with (6.12) concludes the proof. □

Lemma 6.3.7. *Let ℓ, $\mathcal{F}$, $\hat{S}$ and $\hat{f}$ be as defined in Theorem 6.3.5 and let $i \in N$ be such that $\hat{f}(\mathbf{x}_i) > \hat{y}_i$. Then there exists $\tilde{y}_i \in \mathbb{R}$ such that if we define $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, \tilde{y}_i)\}$ and $\tilde{f} = \texttt{ERM}(\mathcal{F}, \ell, \tilde{S})$, then $\tilde{f}(\mathbf{x}_i) < \hat{f}(\mathbf{x}_i)$.*

Proof. Let i be such that $\hat{f}(\mathbf{x}_i) \neq \hat{y}_i$ and assume without loss of generality that $\hat{f}(\mathbf{x}_i) > \hat{y}_i$. Since $\hat{y}_i \in \mathcal{F}(\mathbf{x}_i)$, there exists a function $f' \in \mathcal{F}$ such that $f'(\mathbf{x}_i) = \hat{y}_i$. Now define

$$\phi \;=\; \frac{\hat{\text{risk}}(f', \hat{S}_{-i}) \;-\; \hat{\text{risk}}(\hat{f}, \hat{S}_{-i}) \;+\; 1}{\hat{f}(\mathbf{x}_i) - f'(\mathbf{x}_i)} \quad . \tag{6.13}$$

It holds, by definition, that $\hat{\text{risk}}(f', \hat{S}) > \hat{\text{risk}}(\hat{f}, \hat{S})$ and that $\ell(f'(\mathbf{x}_i), \hat{y}_i) < \ell(\hat{f}(\mathbf{x}_i), \hat{y}_i)$, and therefore the numerator of (6.13) is positive. Furthermore, our assumption implies that the denominator of (6.13) is also positive, so ϕ is positive as well.

Since μ has unbounded subderivatives, there exists $\psi > 0$ large enough such that the subderivative of μ at ψ is greater than ϕ. By the definition of the subderivative, we have that

$$\text{for all } \alpha \geq \psi, \qquad \mu(\psi) + (\alpha - \psi)\phi \;\leq\; \mu(\alpha) \quad . \tag{6.14}$$

Defining $\tilde{y}_i = f'(\mathbf{x}_i) - \psi$ and $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, \tilde{y}_i)\}$, we have that

$$\ell(f'(\mathbf{x}_i), \tilde{y}_i) \;=\; \mu(f'(\mathbf{x}_i) - \tilde{y}_i) \;=\; \mu(\psi) \quad ,$$

and therefore

$$\hat{\text{risk}}(f', \tilde{S}) \;=\; \hat{\text{risk}}(f', \tilde{S}_{-i}) + \ell(f'(\mathbf{x}_i), \tilde{y}_i) \;=\; \hat{\text{risk}}(f', \tilde{S}_{-i}) + \mu(\psi) \quad . \tag{6.15}$$

We further have that

$$\ell(\hat{f}(\mathbf{x}_i), \tilde{y}_i) \;=\; \mu(\hat{f}(\mathbf{x}_i) - \tilde{y}_i) \;=\; \mu(\hat{f}(\mathbf{x}_i) - f'(\mathbf{x}_i) + \psi) \quad .$$

Combining (6.14) with the fact that $\hat{f}(\mathbf{x}_i) - f'(\mathbf{x}_i) > 0$, we get $\mu(\psi) + (\hat{f}(\mathbf{x}_i) - f'(\mathbf{x}_i))\phi$ as a lower bound for the above. Plugging in the definition of ϕ from (6.13), we obtain

$$\ell(\hat{f}(\mathbf{x}_i), \tilde{y}_i) \;\geq\; \mu(\psi) + \hat{\text{risk}}(f', \hat{S}_{-i}) - \hat{\text{risk}}(\hat{f}, \hat{S}_{-i}) + 1 \quad ,$$

and therefore,

$$\hat{\text{risk}}(\hat{f}, \tilde{S}) \;=\; \hat{\text{risk}}(\hat{f}, \tilde{S}_{-i}) + \ell(\hat{f}(\mathbf{x}_i), \tilde{y}_i) \;\geq\; \mu(\psi) + \hat{\text{risk}}(f', \hat{S}_{-i}) + 1 \quad .$$

Comparing the above with (6.15), we get

$$\hat{\text{risk}}(\hat{f}, \tilde{S}) \;>\; \hat{\text{risk}}(f', \tilde{S}) \quad .$$

We now use Lemma 6.3.6 to extend the above to every $f \in \mathcal{F}$ for which $f(\mathbf{x}_i) \geq \hat{f}(\mathbf{x}_i)$, namely, we now have that any such f satisfies $\hat{\text{risk}}(f, \tilde{S}) > \hat{\text{risk}}(f', \tilde{S})$. We conclude that the empirical risk minimizer $\tilde{f}$ must satisfy $\tilde{f}(\mathbf{x}_i) < \hat{f}(\mathbf{x}_i)$. □

Lemma 6.3.8. *Let ℓ and $\mathcal{F}$ be as defined in Theorem 6.3.5, let $S = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^m$ be a dataset, and let $i \in N$ be an arbitrary index. Then the function $g(\tilde{y}) = f(\mathbf{x}_i)$, where $f = \texttt{ERM}(\mathcal{F}, \ell, S_{-i} \cup \{(\mathbf{x}_i, \tilde{y})\})$, is continuous.*

Proof. We first restate ERM as a minimization problem over vectors in $\mathbb{R}^m$. Define the set of feasible values for the points $\mathbf{x}_1, \ldots, \mathbf{x}_m$ to be

$$\mathcal{G} \;=\; \big\{\big(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)\big) \,:\, f \in \mathcal{F}\big\} \quad .$$

Our assumption that $\mathcal{F}$ is a convex set implies that $\mathcal{G}$ is a convex set as well. Now, define the function

$$L(\mathbf{v}, \tilde{y}) \;=\; \ell(v_i, \tilde{y}) + \sum_{j \neq i} \ell(v_j, \hat{y}_j), \qquad \text{where } \mathbf{v} = (v_1, \ldots, v_m) \quad .$$

Finding $f \in \mathcal{F}$ that minimizes the empirical risk with respect to the dataset $S_{-i} \cup \{(\mathbf{x}_i, \tilde{y})\}$ is equivalent to calculating $\min_{\mathbf{v} \in \mathcal{G}} L(\mathbf{v}, \tilde{y})$. Moreover, $g(\tilde{y})$ can be equivalently defined as the value of the i'th coordinate of the vector in $\mathcal{G}$ that minimizes $L(\mathbf{v}, \tilde{y})$.

To prove that g is continuous at an arbitrary point $\tilde{y} \in \mathbb{R}$, we show that for every $\epsilon > 0$ there exists $\delta > 0$ such that if $y \in [\tilde{y} - \delta, \tilde{y} + \delta]$ then $g(y) \in [g(\tilde{y}) - \epsilon, g(\tilde{y}) + \epsilon]$. For this, let $\tilde{y}$ and $\epsilon > 0$ be arbitrary real numbers, and define

$$\mathbf{u} \;=\; \operatorname*{argmin}_{\mathbf{v} \in \mathcal{G}} L(\mathbf{v}, \tilde{y}) \quad .$$

Since ℓ is strictly convex in its first argument, so is L. Consequently, $\mathbf{u}$ is the unique global minimizer of L. Also define

$$\mathcal{G}_\epsilon \;=\; \{\mathbf{v} \in \mathcal{G} \,:\, |v_i - u_i| \geq \epsilon\} \quad .$$

Assume that ϵ is small enough that $\mathcal{G}_\epsilon$ is not empty (if no such ϵ exists, the lemma holds trivially). Note that $\mathbf{u} \notin \mathcal{G}_\epsilon$ for any value of $\epsilon > 0$. Define $\bar{\mathcal{G}}_\epsilon$ to be the closure of $\mathcal{G}_\epsilon$ and let

$$\nu \;=\; \inf_{\mathbf{v} \in \bar{\mathcal{G}}_\epsilon} L(\mathbf{v}, \tilde{y}) - L(\mathbf{u}, \tilde{y}) \quad .$$

Since μ is strictly convex and has unbounded subderivatives, the level-sets of $L(\mathbf{v}, \tilde{y})$, as a function of $\mathbf{v}$, are all bounded. Therefore, there exists $\mathbf{w} \in \bar{\mathcal{G}}_\epsilon$ that attains the infimum above. More precisely, $\mathbf{w}$ is such that $L(\mathbf{w}, \tilde{y}) - L(\mathbf{u}, \tilde{y}) = \nu$. Using uniqueness of the minimizer $\mathbf{u}$, as well as the fact that $\mathbf{w} \neq \mathbf{u}$, we conclude that $\nu > 0$. We have proven that if $\mathbf{v} \in \mathcal{F}$ is such that

$$L(\mathbf{v}, \tilde{y}) \;<\; L(\mathbf{u}, \tilde{y}) \;+\; \nu \quad , \tag{6.16}$$

then $|v_i - u_i| < \epsilon$. It therefore suffices to show that there exists $\delta > 0$ such that if $y \in [\tilde{y} - \delta, \tilde{y} + \delta]$ then the vector $\mathbf{v} \in \mathcal{G}$ that minimizes $L(\mathbf{v}, y)$ satisfies the condition in (6.16).

Since ℓ is convex in its second argument, ℓ is also continuous in its second argument. Thus, there exists $\delta > 0$ such that for all $y \in [\tilde{y} - \delta, \tilde{y} + \delta]$ it holds that both

$$\ell(u_i, \tilde{y}) < \ell(u_i, y) + \nu/2 \quad \text{and} \quad \ell(w_i, y) < \ell(w_i, \tilde{y}) + \nu/2 \quad ,$$

where $\mathbf{w} = \arg\min_{\mathbf{v} \in \mathcal{G}} L(\mathbf{v}, y)$. Therefore,

$$L(\mathbf{u}, \tilde{y}) < L(\mathbf{u}, y) + \nu/2 \quad \text{and} \quad L(\mathbf{w}, \tilde{y}) < L(\mathbf{w}, y) + \nu/2 \quad .$$

Finally, since $\mathbf{w}$ minimizes $L(\mathbf{v}, y)$, we have $L(\mathbf{w}, y) \leq L(\mathbf{u}, y)$. Combining these three inequalities yields the condition in (6.16). □

We are now ready to prove Proposition 6.3.5, and then Theorem 6.3.4.

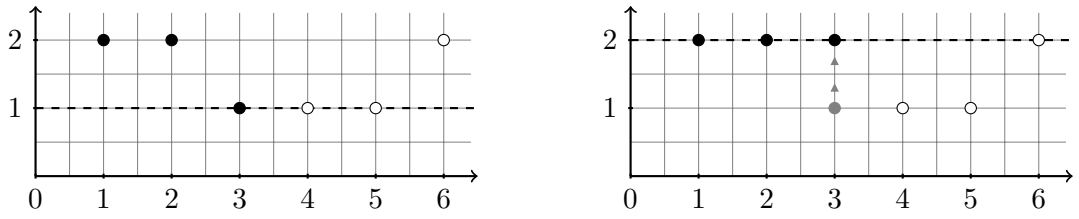
Proof of Proposition 6.3.5. If $\hat{f}(\mathbf{x}_i) = \hat{y}_i$ for all $i \in N$, we are done. Otherwise let i be an index for which $\hat{f}(\mathbf{x}_i) \neq \hat{y}_i$ and assume without loss of generality that $\hat{f}(\mathbf{x}_i) > \hat{y}_i$. Using Lemma 6.3.7, we know that there exists $\tilde{y}_i \in \mathbb{R}$ such that if we define $\tilde{S} = \hat{S}_{-i} \cup \{(\mathbf{x}_i, \tilde{y}_i)\}$ and $f' = \texttt{ERM}(\mathcal{F}, \ell, \tilde{S})$, then f' satisfies $\hat{f}(\mathbf{x}_i) > f'(\mathbf{x}_i)$.

We consider the two possible cases: either $\hat{f}(\mathbf{x}_i) > f'(\mathbf{x}_i) \geq \hat{y}_i$, and therefore $\ell(\hat{f}(\mathbf{x}_i), \hat{y}_i) > \ell(f'(\mathbf{x}_i), \hat{y}_i)$ as required. Otherwise, $\hat{f}(\mathbf{x}_i) > \hat{y}_i > f'(\mathbf{x}_i)$. Using Lemma 6.3.8, we know that $f(\mathbf{x}_i)$ changes continuously with $\tilde{y}_i$, where $f = \texttt{ERM}(\mathcal{F}, \ell, S_{-i} \cup \{(\mathbf{x}_i, \tilde{y}_i)\})$. Relying on the elementary *Intermediate Value Theorem*, we conclude that for some $y \in [\hat{y}_i, \tilde{y}_i]$ it holds that f, the empirical risk minimizer with respect to the dataset $S_{-i} \cup \{(\mathbf{x}_i, y)\})$, satisfies $f(\mathbf{x}_i) = \hat{y}_i$. Once again we have $\ell(\hat{f}(\mathbf{x}_i), \hat{y}_i) > \ell(f(\mathbf{x}_i), \hat{y}_i)$. □

Proof of Theorem 6.3.4. Since $\mathcal{F}$ is not full on X, there are $y_1^*, \ldots, y_n^*$ such that $y_i^* \in \mathcal{F}(\mathbf{x}_i)$ for all i, and $\langle y_1^*, \ldots, y_n^* \rangle \notin \mathcal{F}(X)$. Defining $S = \langle (\mathbf{x}_i, y_i^*) \}_{i=1}^n$, there exists some agent i which isn't satisfied by the output of the ERM algorithm on S. Using Proposition 6.3.5 we conclude that this agent has an incentive to lie. □

It is natural to ask what happens for loss functions that are *sublinear* in the sense that they cannot be bounded from below by any linear function with strictly positive derivative. A property of such loss functions, and the reason why they are rarely used in practice, is that the set of empirical risk minimizers need no longer be convex. It is thus unclear how tie-breaking should be defined in order to find a unique empirical risk minimizer. Furthermore, the following example provides a negative answer to the question of general strategyproofness of ERM with sublinear loss.

Example 6.3.9. We demonstrate that ERM is not strategyproof if $\ell(a, b) = \sqrt{|a - b|}$ and $\mathcal{F}$ is the class of constant functions over $\mathbb{R}$. Let $S = \{(x_1, 1), (x_2, 2), (x_3, 4), (x_4, 6)\}$ and $\tilde{S} = \{(x_1, 1), (x_2, 2), (x_3, 4), (x_4, 4)\}$. Clearly, the local minima of $\hat{\text{risk}}(f, S)$ and $\hat{\text{risk}}(f, \tilde{S})$ have the form $f(x) \equiv y$ where $(x_i, y) \in S$ or $(x_i, y) \in \tilde{S}$, respectively, for some $i \in \{1, 2, 3, 4\}$. The empirical risk minimizer for S is the constant function $f_1(x) \equiv 2$, while that for $\tilde{S}$ is $f_2(x) \equiv 4$. Thus, agent 4 can declare its value to be 4 instead of 6 to decrease its loss from 2 to $\sqrt{2}$.



(a) Truthful dataset. (b) Manipulated dataset.

Figure 6.1: An illustration of Example 6.4.1, which shows that ERM is not strategyproof when agents have uniform distributions over their samples.

6.4 Uniform Distributions Over the Sample

We now turn to settings where a single agent holds a (possibly) nondegenerate distribution over the input space. However, we still do not move to the full level of generality. Rather, we concentrate on a setting where for each agent i, ρ_i is the uniform distribution over the sample points x_{ij}, $j = 1, \ldots, m$. While this setting is equivalent to curve fitting with multiple agents and may be interesting in its own right, we primarily engage in this sort of analysis as a stepping stone in our quest to understand the learning game. The results in this section will function as building blocks for the results of Section 6.5.

Since each agent $i \in N$ now holds a uniform distribution over its sample, we can simply assume that its cost is its average empirical loss on the sample, $\hat{\text{risk}}(\tilde{f}, S_i) = 1/m \sum_{j=1}^{m} \ell(\tilde{f}(\mathbf{x}_{ij}), y_{ij})$. The mechanism's goal is to minimize $\hat{\text{risk}}(\tilde{f}, S)$. We stress at this point that the results in this section also hold if the the samples of the agents differ in size. This is of course true for the negative results, but also holds for the positive ones. As we move to this more general setting, truthfulness of ERM immediately becomes a thorny issue even under absolute loss. Indeed, the next example indicates that more sophisticated mechanisms must be used to achieve strategyproofness.

Example 6.4.1. Let $\mathcal{F}$ be the class of constant functions over $\mathbb{R}^k$, $N = \{1, 2\}$, and assume the absolute loss function is used. Let $S_1 = \{(1, 2), (2, 2), (3, 1)\}$ and $S_2 = \{(4, 1), (5, 1), (6, 2)\}$. The global empirical risk minimizer (according to our tie-breaking rule) is the constant function $f_1(x) \equiv 1$ with $\hat{\text{risk}}(f_1, S_1) = 2/3$. However, if agent 1 declares $\tilde{S}_1 = \{(1, 2), (2, 2), (3, 2)\}$, then the empirical risk minimizer becomes $f_2(x) \equiv 2$, which is the optimal fit for agent 1 with $\hat{\text{risk}}(f_2, S_1) = 1/3$. Figure 6.1 illustrates this example.

6.4.1 Mechanisms with Payments

One possibility to overcome the issue that became manifest in Example 6.4.1 is to consider mechanisms that not only return an allocation, but can also transfer payments to and from the agents based on the inputs they provide. A famous example for such a payment rule is the Vickrey-Clarke-Groves (VCG) mechanism [146, 25, 62]. This mechanism starts from an efficient allocation and computes each agent's payment according to the utility of the other agents, thus aligning the individual interests of each agent with that of society.

In our setting, where social welfare equals the total empirical risk, ERM generates a function, or outcome, that maximizes social welfare and can therefore be directly augmented with VCG

payments. Given an outcome $\hat{f}$, each agent i has to pay an amount of $\hat{\text{risk}}(\hat{f}, \tilde{S}_{-i})$. In turn, the agent can receive some amount $h_i(\tilde{S}_{-i})$ that does *not* depend on the values it has reported, but possibly on the values reported by the other agents. It is well known [62], and also easily verified, that this family of mechanisms is strategyproof: no agent is motivated to lie regardless of the other agents' actions. Furthermore, this result holds for any loss function, and may thus be an excellent solution for some settings.

In many other settings, however, especially in the world of the Internet, transferring payments to and from users can pose serious problems, up to the extent that it might become completely infeasible. The practicality of VCG payments in particular has recently also been disputed for various other reasons [133]. Perhaps most relevant to our work is the fact that VCG mechanisms are in general susceptible to manipulation by coalitions of agents and thus not group strategyproof. It is therefore worthwhile to explore which results can be obtained when payments are disallowed. This will be the subject of the following section.

6.4.2 Mechanisms without Payments

In this section, we restrict ourselves to the absolute loss function. When ERM is used, and for the special case covered in Section 6.3, this function was shown to possess incentive properties far superior to any other loss function. This fuels hope that similar strategyproofness results can be obtained with uniform distributions over the samples, even when payments are disallowed. This does not necessarily mean that good mechanisms without payments cannot be designed for other loss functions, even in the more general setting of this section. We leave the study of such mechanisms for future work.

ERM is *efficient*, i.e. it minimizes the overall loss and maximizes social welfare. In light of Example 6.4.1, we shall now sacrifice efficiency for strategyproofness. More precisely, we seek strategyproof or group strategyproof mechanisms which are at the same time *approximately efficient*. We should stress that the reason we resort to approximation is not to make the mechanism computationally tractable, but to achieve strategyproofness without payments, like we had in Section 6.3.

Example 6.4.1, despite its simplicity, is surprisingly robust against many conceivably truthful mechanisms. The reader may have noticed, however, that the values of the agents in this example are not "individually realizable": in particular, there is no constant function which *realizes* agent 1's values, i.e. fits them with a loss of zero. In fact, agent 1 benefits from revealing values which are consistent with its individual empirical risk minimizer. This insight leads us to design the following simple but useful mechanism, which we will term "project-and-fit":

Input: A hypothesis class $\mathcal{F}$ and a sample $S = \uplus S_i$, $S_i \subseteq \mathcal{X} \times \mathbb{R}$
Output: A function $f \in \mathcal{F}$.
Mechanism:

1. For each $i \in N$, let $f_i = \texttt{ERM}(\mathcal{F}, S_i)$.
2. Define $\tilde{S}_i = \{(\mathbf{x}_{i1}, f_i(x_{i1})), \ldots, (\mathbf{x}_{im}, f_i(x_{im}))\}$.
3. Return $f = \texttt{ERM}(\tilde{S})$, where $\tilde{S} = \uplus_{i=1}^{n} \tilde{S}_i$.

In other words, the mechanism calculates the individual empirical risk minimizer for each agent and uses it to relabel the agent's sample. Then, the relabeled samples are combined, and ERM

is performed. It is immediately evident that this mechanism achieves group strategyproofness at least with respect to Example 6.4.1.

More generally, it can be shown that the mechanism is group strategyproof when $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$. Indeed, it is natural to view our setting through the eyes of voting theory: agents entertain (weak) preferences over a set of alternatives, i.e. the functions in $\mathcal{F}$. In the case of constant functions, agents' preferences are what is known as *single-plateau* [101]: each agent has an interval of ideal points minimizing its individual empirical risk, and moving away from this plateau in either direction strictly decreases the agent's utility. More formally, let a_1, a_2 be constants such that the constant function $f(x) \equiv a$ minimizes an agent's empirical risk if and only if $a \in [a_1, a_2]$. If a_3 and a_4 satisfy $a_3 < a_4 \leq a_1$ or $a_3 > a_4 \geq a_2$, then the agent strictly prefers the constant function a_4 to the constant function a_3. As such, single-plateau preferences generalize the class of single-peaked preferences. For dealing with single-plateau preferences, Moulin [101] defines the class of generalized Condorcet winner choice functions, and shows that these are group strategyproof.

When $\mathcal{F}$ is the class of constant functions and ℓ is the absolute loss, the constant function equal to a median value in a sample S minimizes the empirical risk with respect to S. This is because there must be at least as many values below the median value as are above, and thus moving the fit upward (or downward) must monotonically increase the sum of distances to the values. Via tie-breaking, project-and-fit essentially turns the single-plateau preferences into single-peaked ones, and then chooses the median peak. Once again, group strategyproofness follows from the fact that an agent can only change the mechanism's output by increasing its distance from its own empirical risk minimizer.

Quite surprisingly, project-and-fit is not only truthful but also provides a constant approximation ratio when $\mathcal{F}$ is the class of constant functions or the class of homogeneous linear functions over $\mathbb{R}$, i.e. functions of the form $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$. The class of homogeneous linear functions, in particular, is important in machine learning, for instance in the context of Support Vector Machines [138].

Theorem 6.4.2. *Assume that $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$, $k \in \mathbb{N}$, or the class of homogeneous linear functions over $\mathbb{R}$. Then project-and-fit is group strategyproof and 3-efficient.*

The proof of Theorem 6.4.2 is delegated to Appendix C.1. A simple example shows that the 3-efficiency analysis given in the proof is tight. We generalize this observation by proving that, for the class of constant or homogeneous linear functions and irrespective of the dimension of $\mathcal{X}$, no truthful mechanism without payments can achieve an efficiency ratio better than 3. It should be noted that this lower bound holds for any choice of points x_{ij}. The proof of Theorem 6.4.3 appears in Appendix C.2.

Theorem 6.4.3. *Let $\mathcal{F}$ be the class of constant functions over $\mathbb{R}^k$ or the class of homogeneous linear functions over $\mathbb{R}^k$, $k \in \mathbb{N}$. Then there exists* no *strategyproof mechanism without payments that is $(3-\epsilon)$-efficient for any $\epsilon > 0$, even when $|N| = 2$.*

Let us recapitulate. We have found a group strategyproof and 3-efficient mechanism for the class of constant functions over $\mathbb{R}^k$ and for the class of homogeneous linear functions over $\mathbb{R}$. A matching lower bound, which also applies to multi-dimensional homogeneous linear functions, shows that this result cannot be improved upon for these classes. It is natural to ask at this point if project-and-fit remains strategyproof when considering more complex hypothesis classes, such as homogeneous

linear functions over $\mathbb{R}^k$, $k \geq 2$, or linear functions. An example serves to answer this question in the negative.

Example 6.4.4. We demonstrate that project-and-fit is not strategyproof when $\mathcal{F}$ is the class of linear functions over $\mathbb{R}$. Let $S_1 = \{(0,0),(4,1)\}$ and $S_2 = \{(1,1),(2,0)\}$. Since S_1 and S_2 are individually realizable, the mechanism simply returns the empirical risk minimizer, which is $f(x) = x/4$ (this can be determined by solving a linear program). It further holds that $\hat{\text{risk}}(f, S_2) = 5/8$. If, however, one considers $\tilde{S}_2 = \{(1,1),(2,1)\}$ and the same S_1, then the mechanism returns $\tilde{f}(x) = 1$. Agent 2 benefits from this lie as $\hat{\text{risk}}(\tilde{f}, S_2) = 1/2$.

It is also possible to extend this example to the case of homogeneous linear functions over $\mathbb{R}^2$ by fixing the second coordinate of all points at 1, i.e. mapping each $x \in \mathbb{R}$ to $\mathbf{x}' = (x,1) \in \mathbb{R}^2$. Indeed, the value of a homogeneous linear function $f(\mathbf{x}) = \langle a, b\rangle \cdot \mathbf{x}$ on the point (x,1) is $ax + b$.

Is there some other mechanism which deals with more complex hypothesis classes and provides a truthful approximation? We conjecture that the answer is "no". Some justification for this conjecture is given in Appendix C.3.

Conjecture 6.4.5. *Let $\mathcal{F}$ be the class of homogeneous linear functions over $\mathbb{R}^k$, $k \geq 2$, and assume that $m = |S_i| \geq 3$. Then any mechanism that is strategyproof (in ex-post Nash equilibrium) and surjective must be a dictatorship.*

Conceivably, dictatorship would be an acceptable solution if it could guarantee approximate efficiency. A simple example shows that unfortunately this is not the case.

Example 6.4.6. Consider the class of homogeneous linear functions over $\mathbb{R}^2$, $N = \{1,2\}$. Let $S_1 = \{(\langle 0,1\rangle, 0), (\langle 0+\epsilon, 1\rangle, 0)\}$ and $S_2 = \{(\langle 1,1\rangle, 1), (\langle 1+\epsilon, 1\rangle, 1)\}$ for some $\epsilon > 0$. Any dictatorship has an empirical risk of $1/2$. On the other hand, the function $f(x_1, x_2) = x_1$ has empirical risk $\epsilon/2$. The efficiency ratio increases arbitrarily as ϵ decreases.

6.5 Arbitrary Distributions Over the Sample

In Section 6.4 we established several positive results in the setting where each agent cares about a uniform distribution on its portion of a global training set. In this section we extend these results to the general regression learning setting defined in Section 6.2. More formally, the extent to which agent $i \in N$ cares about each point in $\mathcal{X}$ will now be determined by the distribution function ρ_i, and agent i controls the labels of a finite set of points sampled according to ρ_i. Our strategy in this section will consist of two steps. First, we want to show that under standard assumptions on the hypothesis class $\mathcal{F}$ and the number m of samples, each agent's empirical risk on the training set S_i estimates its real risk according to ρ_i. Second, we intend to establish that, as a consequence, our strategyproofness results are not significantly weakened when we move to the general setting.

Abstractly, let ρ be a probability distribution on $\mathcal{X}$ and let $\mathcal{G}$ be a class of real-valued functions from $\mathcal{X}$ to $[0, C]$. We would like to prove that for any $\epsilon > 0$ and $\delta > 0$ there exists $m \in \mathbb{N}$ such that, if $X_1, \ldots, X_m$ are sampled i.i.d. according to ρ,

$$\Pr\left(\text{for all } g \in \mathcal{G}, \left|\mathbb{E}_{X\sim\rho}[g(X)] - \frac{1}{m}\sum_{i=1}^{m} g(X_i)\right| \leq \epsilon\right) \geq 1 - \delta. \tag{6.17}$$

To establish this bound, we use standard *uniform convergence* arguments. A specific technique is to show that the hypothesis class $\mathcal{G}$ has bounded complexity. The complexity of $\mathcal{G}$ can be measured in various different ways, for example using the pseudo-dimension [112, 63], an extension of the generalized dimension (defined in Chapter 5) to real-valued hypothesis classes, or the Rademacher complexity [11]. If the pseudo-dimension of $\mathcal{G}$ is bounded by a constant, or if the Rademacher complexity of $\mathcal{G}$ with respect to an m-point sample is $O(\sqrt{m})$, then there indeed exists m such that (6.17) holds.

More formally, assume that the hypothesis class $\mathcal{F}$ has bounded complexity, choose $\epsilon > 0$, $\delta > 0$, and consider a sample S_i of size $m = \Theta(\log(1/\delta)/\epsilon^2)$ drawn i.i.d. from the distribution ρ_i of any agent $i \in N$. Then we have that

$$\Pr\left(\text{for all } f \in \mathcal{F}, \quad \left|\text{risk}_i(f) - \hat{\text{risk}}(f, S_i)\right| \;\leq\; \epsilon\right) \;\geq\; 1 - \delta \quad . \tag{6.18}$$

In particular, we want the events in (6.18) to hold simultaneously for all $i \in N$, i.e.

$$\text{for all } f \in \mathcal{F}, \quad \left|\text{risk}_N(f) - \hat{\text{risk}}(f, S)\right| \;\leq\; \epsilon \quad . \tag{6.19}$$

Using the union bound, this is the case with probability at least $1 - n\delta$.

We now turn to strategyproofness. The following theorem implies that mechanisms which do well in the setting of Section 6.4 are also good, but slightly less so, when arbitrary distributions are allowed. Specifically, given a training set satisfying (6.18) for all agents, a mechanism that is strategyproof in the setting of Section 6.4 becomes ϵ-strategyproof, i.e. no agent can gain more than ϵ by lying, no matter what the other agents do. Analogously, a group strategyproof mechanism for the setting of Section 6.4 becomes ϵ-group strategyproof, i.e. there exists an agent in the coalition that gains less than ϵ. Furthermore, efficiency is preserved up to an additive factor of ϵ. We wish to point out that ϵ-equilibrium is a well-established solution concept, the underlying assumption being that agents would not bother to lie if they were to gain an amount as small as ϵ. This concept is particularly appealing when one recalls that ϵ can be chosen to be arbitrarily small.

Theorem 6.5.1. *Let $\mathcal{F}$ be a hypothesis class, ℓ some loss function, and $S = \uplus S_i$ a training set such that for all $f \in \mathcal{F}$ and $i \in N$, $|risk_i(f) - \hat{risk}(f, S_i)| \leq \epsilon/2$, and $|risk_N(f) - \hat{risk}(f, S)| \leq \epsilon/2$. Let M be a mechanism with or without payments.*

1. *If M is (group) strategyproof under the assumption that each agent's cost is $\hat{risk}(\tilde{f}, S_i)$, then M is ϵ-(group) strategyproof in the general regression setting.*
2. *If M is α-efficient under the assumption that the mechanism's goal is to minimize $\hat{risk}(\tilde{f}, S)$, $M(S) = \tilde{f}$, then $risk_N(\tilde{f}) \leq \alpha \cdot \text{argmin}_{f \in \mathcal{F}} risk_N(f) + \epsilon$.*

Proof sketch. We will only prove the first part of the theorem, and only for (individual) strategyproofness. Group strategyproofness as well as the second part of the theorem follow from similar arguments.

Let $i \in N$, and let $\tilde{u}_i(\tilde{S}_i)$ be the utility of agent i when $\tilde{S}$ is reported and assuming a uniform distribution over S_i. Denoting by $\tilde{f}$ the function returned by M given $\tilde{S}$, we have

$$\tilde{u}_i(\tilde{S}) = -\hat{\text{risk}}(\tilde{f}, S_i) + p_i(\tilde{S}) \quad ,$$

where S_i is the training data of agent i with the true labels set by o_i. If M is a mechanism without payments, p_i is the constant zero function. Since M is strategyproof for the uniform distribution, $\tilde{u}_i(S_i, \tilde{S}_{-i}) \geq \tilde{u}_i(\hat{S}_i, \tilde{S}_{-i})$ holds for all $\hat{S}_i$.

On the other hand, let u_i denote agent i's utility function with respect to distribution ρ_i, i.e.

$$u_i(\tilde{S}) = -\text{risk}_i(\tilde{f}) + p_i(\tilde{S}) \quad ,$$

where $\tilde{f}$ is as above. Then, $|u_i(\tilde{S}) - \tilde{u}_i(\tilde{S})| = |\text{risk}_i(\tilde{f}) - \hat{\text{risk}}(\tilde{f}, S_i)|$. By assumption, this expression is bounded by $\epsilon/2$. Similarly, with respect to i's true values S_i, if $M(S_i, \tilde{S}_{-i}) = \hat{f}$, then

$$|u_i(S_i, \tilde{S}_{-i}) - \tilde{u}_i(S_i, \tilde{S}_{-i})| = |\text{risk}_i(\hat{f}) - \hat{\text{risk}}(\hat{f}, S_i)| \leq \epsilon/2 \quad .$$

It follows that for any $\tilde{S}$,

$$u_i(\tilde{S}) - u_i(S_i, \tilde{S}_{-i}) \leq \left(\tilde{u}_i(\tilde{S}) + \frac{\epsilon}{2}\right) - \left(\tilde{u}_i(S_i, \tilde{S}_{-i}) - \frac{\epsilon}{2}\right) \leq \epsilon \quad .$$

□

As discussed above, the conditions of Theorem 6.5.1 are satisfied with probability $1-\delta$ when $\mathcal{F}$ has bounded dimension and $m = \Theta(\log(1/\delta)/\epsilon^2)$. As the latter expression depends logarithmically on $1/\delta$, the sample size only needs to be increased by an additive factor of $\Theta(\log(n)/\epsilon^2)$ to achieve the stronger requirement of (6.19).

Let us examine how Theorem 6.5.1 applies to our positive results. Since ERM with VCG payments is strategyproof and efficient under uniform distributions over the samples, we obtain ϵ-strategyproofness and efficiency up to an additive factor of ϵ when it is used in the general learning game, i.e. with arbitrary distributions. This holds for any loss function ℓ. The project-and-fit mechanism is ϵ-group strategyproof in the learning game when $\mathcal{F}$ is the class of constant functions or of homogeneous linear functions over $\mathbb{R}$, and 3-efficient up to an additive factor of ϵ. This is true only for the absolute loss function.

6.6 Related Work

Previous work in machine learning has investigated the related problem of learning in the presence of inconsistent and noisy training data, where the noise can be either random [91, 61] or adversarial [76, 20]. Barreno et al. [6] consider a specific situation where machine learning is used as a component of a computer security system, and account for the possibility that the training data is subject to a strategic attack intended to infiltrate the secured system. In contrast to these approaches, we do not attempt to design algorithms that can tolerate noise, but instead focus on designing algorithms that discourage the strategic addition of noise.

Closely related to our work is that of Perote and Perote-Peña [110]. The authors essentially study the setting where each agent controls one point of the input space, in a framework that is not learning-theoretic. In addition, they only consider linear regression, and the input space is restricted to be the real line. For that setting, the authors put forward a class of truthful estimators. Rather than looking at the approximation properties of said estimators, they are instead shown to be Pareto-optimal, i.e. there exist no regression lines that are weakly better for all agents, and strictly better for at least one agent.

Our work is also related to the area of *algorithmic mechanism design*, introduced in the seminal work of Nisan and Ronen [105]. Algorithmic mechanism design studies algorithmic problems in a game-theoretic setting where the different participants cannot be assumed to follow the algorithm

but rather act in a selfish way. It has turned out that the main challenge of algorithmic mechanism design is the inherent incompatibility of generic truthful mechanisms with approximation schemes for hard algorithmic problems. As a consequence, most of the current work in algorithmic mechanism design focuses on dedicated mechanisms for hard problems(see, e.g., [88]). What distinguishes our setting from that of algorithmic mechanism design is the need for *generalization* to achieve globally satisfactory results on the basis of a small number of samples. Due to the dynamic and uncertain nature of the domain, inputs are usually assumed to be drawn from some underlying fixed distribution. In addition, as noted above, many of our results focus on a setting without payments, in stark contrast to the vast majority of work on algorithmic mechanism design.

Subsequent work by Meir et al. [97] has extended our results to the realm of classification. This work focuses on the almost degenerate concept class that contains only two functions: the constant positive function, that labels the entire input space positively, and the constant negative hypothesis. Even with respect to this class the problems are nontrivial; Meir et al. have obtained matching upper and lower bounds in this setting, both for deterministic mechanisms and for randomized mechanisms. For an overview and more information on the relation between the work described in this chapter and the work of Meir et al. [97], the reader is referred to Procaccia [113].

6.7 Discussion

The positive results in this chapter are a rare example of mechanism design without money [136]. Essentially, the results of Section 6.4 hold in a setting where the preferences of the agents are single peaked. One of the main contributions of our work, besides introducing the agenda of studying incentives in machine learning, is the the concept of approximation in mechanism design without money: we sacrifice absolute efficiency in order to obtain strategyproofness.

The results presented in this chapter, together with the results of Meir et al. [97], seem to be merely the tip of the iceberg. First, we still have a very limited understanding of incentives in regression learning and classification. Second, there are many other machine learning models waiting to be explored, e.g., clustering.

Part III

Frequency of Manipulation in Elections

Chapter 7

Junta Distributions

7.1 Introduction

In Section 1.1.1 we have surveyed at length the work on the complexity of manipulation in elections. Recall that in general, the agenda is to circumvent the Gibbard-Satterthwaite Theorem (see Theorem 2.4.2) by appealing to computational complexity arguments. Indeed, given a preference profile, although a manipulation may exist in theory, in practice finding a lie that improves the election's outcome might be a computationally hard problem.

The vast majority of papers in this line of research deal with worst-case complexity of manipulation. Worst-case complexity, however, is not a fully satisfactory barrier against manipulation. What we would really like to know is whether it is possible to design reasonable (from a Social Choice point of view) SCFs such that potential manipulators would *usually*—in typical settings—find it hard to solve the manipulation problem.[1]

The notions and results that we discuss in this chapter were, chronologically, the first attempt to investigate the complexity of manipulation under a typical-case mindset rather than a worst-case one. Ideally, one could hope that some of the prominent SCFs, that are known to be hard to manipulate in certain settings, would be frequently hard to manipulate under typical distributions. The main result of this chapter can be interpreted as implying that this is not the case with respect to coalitional manipulation under Scoring Functions.

To be more precise, let us formulate the coalitional manipulation problem, as introduced in Conitzer et al. [35].

Definition 7.1.1. In the Coalitional Weighted Manipulation (CWM) problem under an SCF f, we are given a set of alternatives A, a set of agents N, and a weight $w_i \in \mathbb{R}_+$ for each agent $i \in N$. We are also given a subset of agents $\bar{N} \subseteq N$, a preference profile $R^{\bar{N}}$ for these agents, and a preferred alternative $p \in A$. We are asked whether it is possible to complete $R^{\bar{N}}$ to a profile R^N for all the agents, such that $f(R^N) = p$.

A short discussion is in order. Denote $\hat{N} = N \setminus \bar{N}$. $\bar{N}$ is interpreted as the set of truthful agents, whereas $\hat{N}$ is the set of potential manipulators. The manipulators have complete information about the ballots of the truthful agents, and are trying to coordinate their votes (that is, complete the preference profile) in a way that makes their favorite alternative p win the election.

[1] We do not suggest, however, that worst-case complexity of manipulation is no longer relevant. Worst-case hardness of manipulation is a desirable property in an SCF, and complexity of manipulation it is still one of the prominent tools for comparing different SCFs with respect to their computational properties.

The issue of weights is one we have not discussed before. In this weighted setting, an agent with weight w counts as w agents voting identically. In the context of this chapter, this is well-defined even if w is an arbitrary nonnegative number. Indeed, since the SCFs we discuss are anonymous (indifferent to the identity of the agents) and based on scores, the number of points an agent with weight w awards is simply multiplied by w. For example, under Plurality an agent with weight $1/2$ would award half a point to its favorite alternative. Weighted voting can be justified as relevant in different political or computational settings.

Finally, let us compare the formulation of the problem with the formal definition of manipulation given in Section 2.4. The computational question is not whether the manipulators can change their votes in a way that improves the outcome, but whether they can cast their votes in a way that makes p win. The former formulation seems much more natural in the context of coalitional manipulation, as each manipulator may have different preferences.

Crucially, Conitzer et al. [35] proved the following theorem.

Theorem 7.1.2 (Conitzer et al. [35])**.** *CWM under Borda and Veto is $\mathcal{NP}$-hard, even when the number of alternatives is only 3.*

Our purpose in this chapter is to provide analytical evidence that, despite the theorem, Borda and Veto (an, in general, Scoring Functions) can frequently be manipulated by weighted coalitions in typical settings. The immediate question that comes to mind is: "What are the typical settings?" In other words, which distributions over the instances of CWM should we investigate? In order to tackle these questions, we shall develop a mathematical framework that is based on the concept of *Junta distributions*.

7.2 The Mathematical Framework

Let us first introduce a variation on the CWM problem that we shall study. This version, called SCWM, is especially tailored for Scoring Functions, and its analysis is more straightforward. Given an instance of CWM, let $\sigma_a = \sigma_a^0$ be the score of alternative $a \in A$ based on $R^{\bar{N}}$. Given $R^{\hat{N}}$, let σ_a^i be the score of a based on the first i manipulators in $\hat{N}$, according to some fixed enumeration of $\hat{N}$ and ballots for the manipulators. Denote $\bar{n} = |\bar{N}|$, $\hat{n} = |\hat{N}|$; then $\sigma_a^{\hat{n}}$ is the overall score of alternative a.

Definition 7.2.1. In the Scoring Coalitional Weighted Manipulation (SCWM) problem under a scoring function f, we are given a set of alternatives A, and a set of agents N. We are also given a subset of agents $\bar{N} \subseteq N$, a weight $w_i \in \mathbb{R}_+$ for each agent $i \in \hat{N}$, where $\hat{N} = N \setminus \bar{N}$, the total score σ_a the agents in $\bar{N}$ award to each alternative $a \in A$, and a preferred alternative $p \in A$. We are asked whether it is possible find a ballot $R^{\hat{N}}$ such that $\sigma_p^{\hat{n}} > \sigma_a^{\hat{n}}$ for all alternatives $a \neq p$.

So, the main difference between CWM (under scoring functions) and SCWM is that we do not require that there actually exist $R^{\bar{N}}$ that induces σ_a for all a. Our requirement that $\sigma_p^{\hat{n}} > \sigma_a^{\hat{n}}$ for all alternatives $a \in A \setminus \{p\}$ in fact does not limit generality when compared with CWM, as an equivalent assumption is implicitly made in the results about CWM under scoring functions, namely the *unique winner* or *adversarial tie-breaking* assumption.

We describe a distribution over the instances of a problem as a collection of distributions $\mu = \{\mu_n\}_{n \in \mathbb{N}}$, where μ_n is a distribution over the instances x such that $|x| = n$. The major question, when defining a framework that involves frequency of manipulation, is according to which

distribution over the instances of the manipulation problem the frequency should be measured. Our approach is to analyze problems whose instances are distributed with respect to a distribution that focuses on hard instances of the coalitional manipulation problem. Ideally, we would like to define the distribution in a way that if one manages to produce an algorithm that can usually manipulate instances according to this distinguished "difficult" distribution, it would follow that the same algorithm would usually succeed when the instances are distributed with respect to other typical distributions. This ideal is very ambitious, and we shall not formally demonstrate that it is achieved, but rather provide some analytical evidence suggesting that it might be plausible.

Definition 7.2.2. Let $\mu = \{\mu_n\}_{n\in\mathbb{N}}$ be a distribution over the possible instances of a decision problem L. μ is a *Junta* distribution if and only if μ has the following properties:

1. Hardness: The restriction of L to μ is the problem whose possible instances are only:

$$\bigcup_{n\in\mathbb{N}} \{x : |x| = n \wedge \mu_n(x) > 0\}.$$

   Deciding this restricted problem is still $\mathcal{NP}$-hard.

2. Balance: There exist a constant $\beta > 1$ and $K \in \mathbb{N}$ such that for all $n \geq K$:

$$\frac{1}{\beta} \leq \Pr_{x\sim\mu_n}[L(X) = \text{"yes"}] \leq 1 - \frac{1}{\beta}.$$

3. Dichotomy: for all n and instances x such that $|x| = n$:

$$\mu_n(x) \geq 2^{-\text{poly}n} \vee \mu_n(x) = 0.$$

Assuming L is specifically the SCWM problem under a scoring function f, we also require:

4. Neutrality: Let $a, b \neq p$ be two alternatives, and $\gamma \in \mathbb{R}$. Then

$$\Pr_{x\sim\mu_n}[\sigma_a = \gamma] = \Pr_{x\sim\mu_n}[\sigma_b = \gamma].$$

5. Refinement: Let x be an instance such that $|x| = n$ and $\mu_n(x) > 0$; if all manipulators $i \in \hat{N}$ voted identically (i.e. $R^i = R^j$ for all $i, j \in \hat{N}$), then p would not be elected.

The name "Junta distribution" comes from the idea that in such a distribution, relatively few "powerful" and difficult instances represent all the other problem instances. Alternatively, our ideal is to have a few problematic distributions (the family of Junta distributions) represent other distributions with respect to frequency of manipulation.

The exact choice of properties is of extreme importance (and may be arguable). We shall briefly explain our choices.

The first three properties are formulated in a general way that applies to any decision problem. Hardness is meant to ensure that the Junta distribution contains hard instances. Balance guarantees that a trivial algorithm that always accepts, or always rejects, has a significant chance of failure. The dichotomy property helps in preventing situations where the distribution gives a (positive but) negligible probability to all the hard instances, and a high probability to several easy instances.

We now examine the properties that are specific to the coalitional manipulation problem. Neutrality focuses the attention on distributions that are natural from a social choice point of view, where no alternative is *a priori* preferred to another. This property is also important from a computational point of view, as instances where some alternatives have significantly higher initial scores than other alternatives are easier to decide.

Finally, refinement is less important than the other four properties, but seems to help in concentrating the probability on hard instances.

We presently introduce the last building blocks of our mathematical framework. The next term is a well-known one in the theory of average-case complexity.

Definition 7.2.3. A *distributional problem* is a pair $\langle L, \mu \rangle$ where L is a decision problem and μ is a distribution over the set $\{0,1\}^*$ of possible inputs.

Informally, an algorithm is a heuristic polynomial time algorithm for a distributional problem if it runs in polynomial time, and fails only on a small fraction of the inputs. We now give a formal definition; this definition is inspired by Trevisan [144].

Definition 7.2.4. Let L be a decision problem and let $\langle L, \mu \rangle$ be a distributional problem. An algorithm ALG is a *deterministic heuristic polynomial time* algorithm for $\langle L, \mu \rangle$ if ALG always runs in polynomial time, and there exists a polynomial q of degree at least 1 and $K \in \mathbb{N}$ such that for all $n \geq K$:

$$\Pr_{x \sim \mu^n}[\text{ALG}(x) \neq L(x)] \leq \frac{1}{q(n)}. \tag{7.1}$$

The following statement we take to be self-evident. Fix some scoring function f, and let ALG be an algorithm for SCWM under f. Now, suppose ALG is a heuristic polynomial time algorithm for SCWM under f with respect to most typical distributions μ over the instances of the problem. Then SCWM under f is frequently tractable.

Unfortunately, showing that an algorithm is a heuristic polynomial time algorithm with respect to "most typical distributions" currently seems out of our reach. We are able, though, to devise an algorithm that is a heuristic polynomial time algorithm for SCWM under f with respect to one distribution, which, incidentally, is a Junta distribution. We suggest that this can be interpreted as evidence that the our algorithm also does well with respect to other typical distributions.

7.3 Formulation, Proof, and Justification of Main Result

Recall that under Borda and Veto, CWM is $\mathcal{NP}$-hard, even with 3 alternatives. We would like to discuss a family of scoring functions that includes Borda and Veto, but does not include, e.g., Plurality.

Definition 7.3.1. Let f be a scoring function with parameters $\alpha = \langle \alpha_1, \ldots, \alpha_m \rangle$. We say that f is *sensitive* iff $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_{m-1} > \alpha_m = 0$ (notice the strict inequality on the right hand side).

Since Borda and Veto are examples of sensitive scoring functions, we would like to know how resistant this family of SCFs is with respect to coalitional manipulation. Our main result is as follows:

Theorem 7.3.2. *Let f be a sensitive scoring function, and assume the number of alternatives m is constant. Then there exists a distribution $\mu^* = \mu^*(f)$ that is a Junta distribution with respect to SCWM under f, and an algorithm that is a heuristic polynomial time algorithm for SCWM under f with respect to μ^*.*

Intuitively, the instances of SCWM that are hard are those that require a very specific partitioning of the agents in $\hat{N}$ to subsets, where each subset votes unanimously. These instances are rare under any typical distribution; this insight will ultimately yield the theorem.

The following proposition generalizes theorems of Conitzer et al. [35] regarding Borda and Veto, and justifies our focus on the family of sensitive scoring functions. A stronger version of Proposition 7.3.3 has been independently proven by Hemaspaandra and Hemaspaandra [65]. Nevertheless, we include our proof, since it will be required in proving the hardness property of the Junta distribution we shall design.

Proposition 7.3.3. *Let f be a sensitive scoring function. Then CWM under f is $\mathcal{NP}$-hard, even with 3 alternatives.*

The proof will require:

Definition 7.3.4. In the PARTITION problem, we are given a set of integers $\{k_i\}_{i\in\{1,\ldots,t\}}$, summing to $2K$, and are asked whether a subset of these integers sum to K.

It is well-known that PARTITION is $\mathcal{NP}$-complete.

Proof of Proposition 7.3.3. We reduce an arbitrary instance of PARTITION to the following CWM instance. There are 3 alternatives, a, b, and p. In $\bar{N}$, there are $K(4\alpha_1 - 2\alpha_2) - 1$ agents voting aR^jbR^jp, and $K(4\alpha_1 - 2\alpha_2) - 1$ agents voting bR^jaR^jp. In $\hat{N}$, for every k_i there is an agent i with weight $2(\alpha_1 + \alpha_2)k_i$. Observe that from $\bar{N}$, both a and b get $(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2)$ points.

Assume first that a partition exists. Let the agents i in $\hat{N}$ in one half of the partition vote pR^iaR^ib, and let the other half vote pR^ibR^ia. By this vote, a and b each have

$$(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2) + 2K(\alpha_1 + \alpha_2)\alpha_2 = (\alpha_1 + \alpha_2)(4K\alpha_1 - 1)$$

points, while p has $(\alpha_1 + \alpha_2)4K\alpha_1$ points; thus there is a manipulation.

Conversely, assume that a manipulation exists. Clearly there must exist a manipulation where all the agents in $\hat{N}$ vote either pR^iaR^ib or pR^ibR^ia, because the manipulators do not gain anything by not placing p at the top under a scoring function. In this ballot, p has $(\alpha_1 + \alpha_2)4K\alpha_1$ points, while a and b already have $(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2)$ points from $\bar{N}$. Therefore, a and b must gain less than $(2\alpha_2K + 1)(\alpha_1 + \alpha_2)$ points from the agents in $\hat{N}$. Each agent corresponding to k_i contributes $2(\alpha_1 + \alpha_2)\alpha_2k_i$ points; it follows that the sum of the k_i corresponding to the agents voting pR^iaR^ib is less than $K + \frac{1}{2\alpha_2}$, and likewise for the agents voting pR^ibR^ia. Equivalently, the sum can be at most K, since all k_i are integers and we can assume without loss of generality that $\alpha_2 \geq 1$. In both cases the sum must be at most K; hence, this is a partition. □

Since an instance of CWM can be translated into an instance of SCWM in the obvious way, we have:

Corollary 7.3.5. *Let f be a sensitive scoring function. Then SCWM under f is $\mathcal{NP}$-hard, even with 3 alternatives.*

7.3.1 A Junta Distribution

For ease of exposition, we slightly abuse notation from this point in the chapter onwards, denoting $n = \hat{n} = |\hat{N}|$ (rather than $n = |N|$), and assuming the number of alternatives is $m+1$ rather than m (so $|A \setminus \{p\}| = m$). Further, we assume f is a sensitive scoring function, and denote $W = \sum_{i \in \hat{N}} w_i$.

Consider a distribution $\mu^*(f) = \mu^* = \{\mu^*_n\}_{n \in \mathbb{N}}$ over the instances of SCWM in f where each μ^*_n is induced by the following sampling algorithm:

1. Fix a polynomial $q = q(n)$.
2. $\forall i \in \hat{N}$: Randomly and independently choose $w_i \in [0, 1]$ (up to $O(n)$ bits of precision, i.e., in intervals of $1/2^{q(n)}$).
3. $\forall a \in A \setminus \{p\}$: Randomly and independently choose $\sigma_a \in [(\alpha_1 - \alpha_2)W, \alpha_1 W]$ (up to $O(n)$ bits of precision).

Remark 7.3.6. Although the distribution is in fact discrete — the weights, for example, are uniformly distributed in $\{0, 1/2^{q(n)}, 2/2^{q(n)}, 3/2^{q(n)}, \ldots, 1\}$ — we treat it below as continuous for the sake of clarity.

We assume that $\sigma_p = 0$, i.e., all agents in $\bar{N}$ rank p last. This assumption does not limit generality. If it holds for an alternative a that $\sigma_a \leq \sigma_p$, then alternative a will surely lose, since the manipulators all rank p first. Therefore, if $\sigma_p > 0$, we may simply normalize the scores by subtracting σ_p from the scores of all alternatives. This is equivalent to our assumption.

Remark 7.3.7. We feel that μ^* is perhaps the natural distribution with respect to which coalitional manipulation in scoring functions should be studied. Even if one disagrees with the exact definition of a Junta distribution, μ^* should still satisfy many reasonable conditions one could produce.

We shall, of course, (presently) prove that the distribution possesses the properties of a Junta distribution.

Lemma 7.3.8. *Let f be a sensitive scoring function, and assume m is constant. Then μ^* is a Junta distribution with respect to SCWM under f.*

Before proving the proposition, let us formulate a basic result from probability theory that we shall require. Informally, the lemma states that the average of independent identically distributed (i.i.d.) random variables is almost always close to the expectation.

Lemma 7.3.9 (Chernoff's Bounds [3])**.** *Let $X_1, \ldots, X_t$ be i.i.d. random variables such that $\beta \leq X_i \leq \gamma$ and $\mathbb{E}[X_i] = \nu$. Then for any $\epsilon > 0$, it holds that:*

1. $\Pr[\frac{1}{t}\sum_{i=1}^{t} X_i \geq \nu + \epsilon] \leq e^{-2t\frac{\epsilon^2}{(\gamma-\beta)^2}}$
2. $\Pr[\frac{1}{t}\sum_{i=1}^{t} X_i \leq \nu - \epsilon] \leq e^{-2t\frac{\epsilon^2}{(\gamma-\beta)^2}}$,

where e is the base of the natural logarithm.

Proof of Lemma 7.3.8. We first observe that neutrality is obviously satisfied, and dichotomy holds by Remark 7.3.6.

The proof of the Hardness property relies on the reduction from PARTITION in Proposition 7.3.3. The reduction generates instances x of CWM in f with 3 alternatives, where $W = 4(\alpha_1 + \alpha_2)K$, and

$$\sigma_a = \sigma_b = (K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2) = (\alpha_1 - \alpha_2/2)W - (\alpha_1 + \alpha_2),$$

for some K that originates in the PARTITION instance. These instances satisfy

$$(\alpha_1 - \alpha_2)W \leq \sigma_a, \sigma_b \leq \alpha_1 W \quad .$$

It follows that $\mu^*(x) > 0$ (after scaling down the weights).[2]

We now prove that μ^* has the balance property. If for all $a \in A \setminus \{p\}$, $\sigma_a > (\alpha_1 - \alpha_2/m)W$, then clearly there is no manipulation, since at least $\alpha_2 W$ points are given by the agents in $\hat{N}$ to the undesirable alternatives $A \setminus \{p\}$. This happens with probability at least $\frac{1}{m^m}$.

On the other hand, consider the situation where for all $a \in A \setminus \{p\}$,

$$\sigma_a < \left(\alpha_1 - \frac{m^2 - 1}{m^2}\alpha_2\right) W; \tag{7.2}$$

this occurs with probability at least $\frac{1}{(m^2)^m}$. Intuitively, if the manipulators could distribute their votes in such a way that each $a \in A \setminus \{p\}$ is ranked last in exactly $1/m$-fraction of the votes, this would be a successful manipulation: each $a \in A \setminus \{p\}$ would gain at most an additional $\frac{m-1}{m}\alpha_2 W$ points. Unfortunately, this is usually not the case, but the following condition is sufficient for a successful manipulation (assuming Condition (7.2) holds). Partition the manipulators to m disjoint subsets $\hat{N}_1, \ldots, \hat{N}_m$ (w.l.o.g. of size n/m), and denote by W_j the total weight of the votes in $\hat{N}_j$. The condition is that for all $j \in \{1, \ldots, m\}$:

$$(1 - 1/m) \cdot 1/2 \cdot n/m \leq W_j \leq (1 + 1/m) \cdot 1/2 \cdot n/m. \tag{7.3}$$

This condition is sufficient, because if the agents in $\hat{N}_j$ all rank $a \in A \setminus \{p\}$ last, the fraction of the agents in $\hat{N}$ that gives a points is at most:

$$\frac{(m-1)(1+1/m)}{(m-1)(1+1/m) + 1 - 1/m} = \frac{m^2 - 1}{m^2 + m - 2}.$$

Hence the number of points a gains from the manipulators is at most:

$$\frac{m^2 - 1}{m^2 + m - 2}\alpha_2 W \leq \frac{m^2 - 1}{m^2}\alpha_2 W < \alpha_1 W - \sigma_a.$$

Furthermore, by Lemma 7.3.9 and the fact that the expected total weight of n/m agents is $1/2 \cdot n/m$, the probability that Condition (7.3) holds is at least $1 - 2e^{-\frac{2n}{m^3}}$. Since m is a constant, this probability is larger than $1/2$ for a large enough n.

Finally, it can easily be seen that μ^* has the refinement property: if all manipulators rank p first and $a \in A \setminus \{p\}$ second, then p gets $\alpha_1 W$ points, and a gets $\alpha_2 W + \sigma_a$ points. But $\sigma_a \geq (\alpha_1 - \alpha_2)W$, hence $\sigma_p^n \leq \sigma_a^n$. □

[2] It seems the reduction can be generalized for a larger number of alternatives. The hard instances are the ones where all undesirable alternatives but two have approximately $(\alpha_1 - \alpha_2)W$ initial points, and two problematic alternatives have approximately $(\alpha_1 - \alpha_m/2)W$ points. These instances have a positive probability under μ^*.

Algorithm 7.3.1 Decides SCWM

```
1: procedure GREEDY(σ, w, p)
2:     for all a ∈ A do                                   ▷ Initialization
3:         σ_a^0 ← σ_a
4:     end for
5:     for i = 1 to n do                                  ▷ All agents in N̂
6:         Sort A \ {p} by σ^{i−1}: σ_{a_1}^{i−1} ≤ σ_{a_2}^{i−1} ≤ ··· σ_{a_m}^{i−1}
7:         agent i votes pR^i a_1 R^i a_2 R^i ··· R^i a_m
8:         for j = 1 to m do                              ▷ Update score
9:             σ_{a_j}^i ← σ_{a_j}^{i−1} + w_i α_{j+1}
10:        end for
11:        σ_p^i ← σ_p^{i−1} + w_i α_1
12:    end for
13:    if argmax_{a∈A} σ_a^n = {p} then                   ▷ p wins
14:        return true
15:    else
16:        return false
17:    end if
18: end procedure
```

7.3.2 A Heuristic Polynomial Time Algorithm

We now present our greedy algorithm for SCWM under scoring functions. The algorithm is imaginitively called GREEDY, and given as Algorithm 7.3.1. We enumerate the agents in $\hat{N}$ by setting $\hat{N} = \{1, \ldots, n\}$, and we denote their weights by $\mathbf{w} = \langle w_1, \ldots, w_n \rangle$, and their initial scores (based on $\bar{N}$) by $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$.

The agents in $\hat{N}$, according to their given order, each rank p first, and the rest of the alternatives in an order inversely proportional to their current score: the alternative with lowest score is ranked second, the alternative with second lowest score is ranked third, and so on. GREEDY accepts if and only if p wins under this ballot.

This algorithm, designed specifically for scoring functions, is a realization of an abstract greedy algorithm: at each stage, agent i ranks the alternatives in $A \setminus \{p\}$ in an order that minimizes the highest score that any $a \in A \setminus \{p\}$ obtains after the current vote. If there is a tie among several permutations, the agent chooses the option such that the second highest score is as low as possible, etc. In any case, every manipulator always ranks p first. In fact, GREEDY can be considered a generalization of the greedy algorithm given by Bartholdi et al. [8].

We now set our sights on proving that Algorithm 7.3.1 is a heuristic polynomial time algorithm for SCWM under sensitive scoring functions with respect to μ^*.

Lemma 7.3.10. *If there exists $i_0 \in \hat{N}$, during the execution of* GREEDY*, and two distinct alternatives $a, b \in A \setminus \{p\}$ such that*

$$|\sigma_a^{i_0} - \sigma_b^{i_0}| \leq \alpha_2, \tag{7.4}$$

then for all $i \geq i_0$ it holds that $|\sigma_a^i - \sigma_b^i| \leq \alpha_2$.

Proof. The proof is by induction on i. The base of the induction is given by equation (7.4). Assume that $|\sigma_a^i - \sigma_b^i| \leq \alpha_2$, and without loss of generality $\sigma_a^i \geq \sigma_b^i$. By the algorithm, agent $i+1$ ranks b

higher than a, and therefore:

$$\sigma_b^{i+1} - \sigma_a^{i+1} \geq -\alpha_2. \tag{7.5}$$

Since p is always ranked first, and the weight of each agent is at most 1, b gains at most α_2 points. Therefore:

$$\sigma_b^{i+1} - \sigma_a^{i+1} \leq \alpha_2. \tag{7.6}$$

Combining equations (7.5) and (7.6) completes the proof. □

Lemma 7.3.11. *Let $a, b \in A \setminus \{p\}$ be two distinct alternatives, and suppose that there exists $i_0 \in \hat{N}$ such that $\sigma_a^{i_0} \geq \sigma_b^{i_0}$, and $i_1 \geq i_0$ such that $\sigma_b^{i_1} \geq \sigma_a^{i_1}$. Then for all $i \geq i_1$ it holds that $|\sigma_a^i - \sigma_b^i| \leq \alpha_2$.*

Proof. Assume that there exists i_0 such that $\sigma_a^{i_0} \geq \sigma_b^{i_0}$, and $i_1 \geq i_0$ such that $\sigma_b^{i_1} \geq \sigma_a^{i_1}$; w.l.o.g. $i_1 > i_0$ (otherwise at stage i_0 it holds that $\sigma_b^{i_0} = \sigma_a^{i_0}$, and then we finish by Lemma 7.3.10). Then there must be $i_2 \in \hat{N}$ such that $i_0 \leq i_2 < i_1$ and $\sigma_a^{i_2} \geq \sigma_b^{i_2}$ but $\sigma_b^{i_2+1} \geq \sigma_a^{i_2+1}$. Since the weight of each agent is at most 1, b gains at most α_2 points from agent $i_2 + 1$. Hence the conditions of Lemma 7.3.10 hold for i_2, which implies that for all $i \geq i_2$: $|\sigma_a^i - \sigma_b^i| \leq \alpha_2$. In particular $i_1 \geq i_2$, hence the lemma follows. □

Lemma 7.3.12. *Let f be a sensitive scoring function, and assume* Greedy *errs on a "yes" instance of SCWM under f, i.e.* Greedy *returns* ***false****. Then there is $d \in \{2, \ldots, m\}$, and a subset of alternatives $D = \{a_1, \ldots, a_d\}$, such that:*

$$\sum_{j=1}^{d} (\alpha_1 W - \sigma_{a_j}) - \sum_{j=1}^{d-1} (j \cdot \alpha_2) \leq W \sum_{j=1}^{d} \alpha_{m+2-j} \leq \sum_{j=1}^{d} (\alpha_1 W - \sigma_{a_j}). \tag{7.7}$$

Proof. For the inequality on the right hand side, for any d alternatives, even if all agents in $\hat{N}$ rank them last in every vote, the total points distributed among them is $W \sum_{j=1}^{d} \alpha_{m+2-j}$. Suppose for contradiction that this inequality does not hold, then there must be some alternative a_j that gains at least $\alpha_1 W - \sigma_{a_j}$ points from the manipulators, implying that this alternative has at least $\alpha_1 W$ points. However, p also has at most $\alpha_1 W$ points, and we assumed that there is a successful manipulation; this is a contradiction.

For the inequality on the left hand side, assume the algorithm erred. Then for $i_1 \in \hat{N}$, there is an alternative a_1 such that $\sigma_{a_1}^{i_1} \geq \alpha_1 W$ (w.l.o.g. only one alternative passes this threshold simultaneously). Denote $\hat{N}' = \{1, \ldots, i_1\}$, and let W' be the total weight of the agents in $\hat{N}'$. Agent i_1 did not rank a_1 last, since $\alpha_{m+1} = 0$, and thus ranking an alternative last gives it no points. We have that there is another alternative a_2 such that: $\sigma_{a_2}^{i_0-1} \geq \sigma_{a_1}^{i_0-1}$. By Lemma 7.3.11, $\sigma_{a_1}^{i_0} - \sigma_{a_2}^{i_0} \leq \alpha_2$, and thus $\sigma_{a_2}^{i_0} \geq \alpha_1 W - \alpha_2$. If these alternatives were not always ranked last by the agents of $\hat{N}'$, there must be another alternative a_3 who was ranked strictly higher by some agent in $\hat{N}'$, w.l.o.g. higher than a_2. Therefore, we have from Lemma 7.3.11 that: $\sigma_{a_2}^{i_0} - \sigma_{a_3}^{i_0} \leq \alpha_2$, and so a_3 has a total of at least $\alpha_1 W - 2\alpha_2$ points.

By inductively continuing the above reasoning, we obtain a subset D of d alternatives (possibly $d = m$), who were always ranked in the d last positions by the agents in $\hat{N}'$, and for a_l it holds that: $\sigma_{a_l}^{i_0} \geq \alpha_1 W - (l-1)\alpha_2$. Therefore, the total points a_l gained from $\hat{N}'$ is at least $\alpha_1 W - l\alpha_2 - \sigma_{a_l}$. Since the total points distributed by the agents in $\hat{N}'$ to the d last alternatives is $W' \sum_{j=1}^{d} \alpha_{m+2-j}$, we have:

$$\sum_{j=1}^{d} (\alpha_1 W - \sigma_{a_j}) - \sum_{j=1}^{d-1} (j \cdot \alpha_2) \leq W' \sum_{j=1}^{d} \alpha_{m+2-j} \leq W \sum_{j=1}^{d} \alpha_{m+2-j} \quad ,$$

which yields the left inequality in the formulation of the lemma. □

Lemma 7.3.13. *Let f be a sensitive scoring function and let m be a constant. Then* Greedy *is a deterministic heuristic polynomial time algorithm for SCWM under f with respect to μ^*.*

Proof. It is obvious that if the given instance has no successful manipulation, then the greedy algorithm would indeed answer that there is no manipulation, since the algorithm is constructive (it actually selects specific votes for the manipulators).

We wish to bound the probability that there is a manipulation and the algorithm erred. By Lemma 7.3.12, a necessary condition for this to occur is as specified in Equation (7.7), or equivalently:

$$W\sum_{j=1}^{d}\alpha_1 - W\sum_{j=1}^{d}\alpha_{m+2-j} - \frac{d(d-1)}{2}\alpha_2 \leq \sum_{j=1}^{d}\sigma_{a_j} \leq W\sum_{j=1}^{d}\alpha_1 - W\sum_{j=1}^{d}\alpha_{m+2-j}. \tag{7.8}$$

In this case the algorithm may err; but what is the probability of (7.8) holding? Fix a subset $D = \{a_1, \ldots, a_d\} \subseteq A$ of size $d \in \{2, \ldots, m\}$. $\sum_{j=1}^{d}\sigma_{a_j}$ is a random variable that takes values in $[d(\alpha_1 - \alpha_2)W, d\alpha_1 W]$. By conditioning on the values of σ_{a_j}, $j = 1, \ldots, d-1$, we have that the probability of $\sum_{j=1}^{d}\sigma_{a_j}$ taking values in some interval $[\beta, \gamma]$ is at most the probability of σ_{a_d} taking a value in an interval of size $\gamma - \beta$, which is at most $\frac{\gamma-\beta}{\alpha_1 W - (\alpha_1-\alpha_2)W}$, since σ_{a_d} is uniformly distributed. By Lemma 7.3.9, $W < n/4$ with probability at most $\epsilon(n) = e^{-\frac{n}{8}}$. On the other hand, if $W \geq n/4$, then (7.8) holds for D with probability at most

$$\frac{\frac{d(d-1)}{2}\alpha_2}{\alpha_1 W - (\alpha_1-\alpha_2)W} = \frac{d(d-1)}{2W} \leq \frac{2d(d-1)}{n} = \frac{1}{q^D(n)},$$

for some polynomial q^D. We complete the proof by showing that (7.1) holds:

$$\begin{aligned}\Pr_{x\sim\mu_n^*}[\textsc{Greedy}(x) \neq \text{SCWM}(x)] &\leq \Pr[W \geq n/4 \wedge (\exists D \subseteq A \; s.t. \; |D| \geq 2 \wedge (7.8))] + \Pr[W < n/4] \\ &\leq \sum_{D\subseteq A:|D|\geq 2}\frac{1}{q^D(n)} + \epsilon(n) \\ &\leq \frac{1}{\text{poly } n}\end{aligned}$$

The last inequality follows from the assumption that $m = O(1)$. □

Clearly, Theorem 7.3.2 directly follows.

7.3.3 The Greedy Algorithm and the Uniform Distribution

In the previous subsection we have seen that Algorithm 7.3.1 is a heuristic polynomial time algorithm with respect to our Junta distribution μ^*. We have suggested that the algorithm also does well with respect to other distributions. In this subsection we support this claim by showing that Algorithm 7.3.1 is also a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM.

For the sake of consistency with previous results, we shall consider a uniform distribution over votes that may produce unfeasible ballots. Nevertheless, equivalent results can be obtained for

feasible (discrete) distributions over votes, and in fact generalizations of some of these results are obtained in Chapter 8. So, in this section we assume that each truthful agent $i \in \bar{N}$, where $|\bar{N}| = \bar{n}$, awards each alternative $a \in A$, including p, a score independently and uniformly distributed in $[0, \alpha_1]$. Further, we assume that the votes are unweighted; this does not limit the generality of our results, since we use lower bounds that depend only on the total weight of the manipulators in $\hat{N}$ (where, as before, we denote $|\hat{N}| = \hat{n} = n$); the individual weights are of no consequence.

We distinguish between two cases in our results, depending on the ratio between the number of truthful agents $\bar{n}$ and the number of manipulators n:

1. $n/\sqrt{\bar{n}} < 1/q(n)$ for some polynomial q of degree at least 1.
2. $n/\sqrt{\bar{n}} > q(\log n)$ for some polynomial q of degree at least 1.

The middle ground that is not covered by the two cases remains an open problem. Before we tackle the first case, we require a lower bound of sorts on the probability that an instance of SCWM is very easy to decide. Since the manipulators in $\hat{N}$ can award an alternative at most $\alpha_1 n$ points, the manipulators cannot make an alternative a beat another alternative b if $\sigma_b - \sigma_a > \alpha_1 n$. In particular, if for every two alternatives a and b it holds that $|\sigma_a - \sigma_b| > \alpha_1 n$, then the manipulators cannot affect the outcome of the election. Moreover, Algorithm 7.3.1 always decides such an instance correctly: if $\sigma_p < \sigma_a$ for some $a \in A \setminus \{p\}$, then the instance is a "no" instance, and in this case the algorithm never errs; and if $\sigma_p > \sigma_a$ for all $a \in A \setminus \{p\}$, then the instance is a "yes" instance, and any vote of the manipulators is sufficient to make p win. We have obtained the following Lemma:

Lemma 7.3.14. *Consider an instance of SCWM where for all $a, b \in A$, $|\sigma_a - \sigma_b| > \alpha_1 n$. Then the instance is a "yes" instance iff $\sigma_p > \sigma_a$ for all alternatives $a \in A \setminus \{p\}$, and the instance is correctly decided by Algorithm 7.3.1.*

This Lemma, together with the Central Limit Theorem, yields the first result.

Proposition 7.3.15. *Let the number of alternatives m be a constant. Then Algorithm 7.3.1 is a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM which satisfy $n/\sqrt{\bar{n}} < 1/q(n)$ for some polynomial $q(n)$ of degree at least 1.*

The proof of this proposition, given Lemma 7.3.14, is basically a special case of Theorem 8.2.3 given in Chapter 8, and thus is omitted at this point.

Moving on to the second case, we require the following lemma, which is not superceded by the results of Section 8:

Lemma 7.3.16. *Let $\epsilon = \frac{\alpha_2}{2(m+1)}$, and consider an instance of SCWM where for all $a, b \in A$, $|\sigma_a - \sigma_b| < \epsilon n$. Then this instance is a "yes" instance, and is correctly decided by Algorithm 7.3.1.*

Proof. Obviously, it is sufficient to prove that the algorithm constructively finds a successful ballot that makes p win. Let $A' \subseteq A \setminus \{p\}$ be the set of undesirable alternatives that had maximal score among the alternatives in $A \setminus \{p\}$ at some stage during the execution of the algorithm, where by stage we mean an iteration of the while loop in lines 5–12. Formally:

$$A' = \{b \in A \setminus \{p\} : \ \exists i \in \{0, \ldots, n-1\} \ s.t. \ \sigma_b^i = \max_a \sigma_a^i\}.$$

By the algorithm, at any stage some alternative from A' is ranked last by an agent in $\hat{N}$, i.e., is given 0 points; the other alternatives in A' receive at any stage at most α_2 points. Therefore, the

total number of points the alternatives in A' receive from the manipulators is at most $(d-1)\alpha_2 n$, where $|A'| = d$. Consequently, if σ_a^n is the score of alternative a when the algorithm terminates,

$$\sum_{a \in A'} \sigma_a^n \leq \sum_{a \in A'} \sigma_a + (d-1)\alpha_2 n.$$

Let $a_0^* \in \text{argmax}_{a \in A'} \sigma_a$, and $a_1^* \in \text{argmax}_{a \in A'} \sigma_a^n$. By Lemma 7.3.11, when the algorithm terminates it holds that the scores of all alternatives in A' are within α_2 of one another. Therefore:

$$\sigma_{a_1^*}^n \leq \sum_{a \in A'} \sigma_a + (d-1)\alpha_2 n - \sum_{a_1^* \neq a \in A'} \sigma_a^n \leq d\sigma_{a_0^*}^n + (d-1)\alpha_2 n - (d-1)(\sigma_{a_1^*}^n - \alpha_2).$$

Through some algebraic manipulations, we obtain:

$$\sigma_{a_1^*}^n \leq \sigma_{a_0^*} + n\left(\frac{d-1}{d}\alpha_2\right) + \frac{d-1}{d}\alpha_2 \leq \sigma_{a_0^*} + n\left(\frac{m}{m+1}\alpha_2\right) + \frac{m}{m+1}\alpha_2.$$

Now, we have that:

$$\begin{aligned}
\sigma_p^n - \sigma_{a_1^*}^n &\geq (\sigma_p + \alpha_1 n) - \left(\sigma_{a_0^*} + \left(\frac{m}{m+1}\alpha_2\right) n + \frac{m}{m+1}\alpha_2\right) \\
&\geq \alpha_1 n - \frac{\alpha_2}{2(m+1)} n - \left(\frac{m}{m+1}\alpha_2\right) n - \frac{m}{m+1}\alpha_2 \\
&\geq \frac{\alpha_2}{2(m+1)} n - \frac{m}{m+1}\alpha_2 \\
&> 0.
\end{aligned}$$

The second transition follows from the assumption that $\sigma_p \geq \sigma_{a_0^*} - \epsilon n$, the third transition from the fact that $\alpha_1 \geq \alpha_2$, and the last transition holds for a large enough n. □

Proposition 7.3.17. *Let the number of alternatives m be a constant. Then Algorithm 7.3.1 is a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM which satisfy $n/\sqrt{\bar{n}} > q(\log n)$ for some polynomial q of degree at least 1.*

Proof. Let $\epsilon = \frac{\alpha_2}{2(m+1)}$. By Lemma 7.3.16, the probability that the algorithm does not err is at least:

$$\Pr[\forall a, b \in A,\ |\sigma_a - \sigma_b| < \epsilon n] = 1 - \Pr[\exists a, b \in A \ s.t.\ \sigma_a - \sigma_b > \epsilon n].$$

By the union bound:

$$\Pr[\exists a, b \in A \ s.t.\ \sigma_a - \sigma_b > \epsilon n] \leq \sum_{a,b \in A} \Pr[\sigma_a - \sigma_b > \epsilon n].$$

Fix $a, b \in A$, and let X_i be $S_a^i - S_b^i$, where S_a^i is the score given to an alternative a by agent $i \in \bar{N}$ (as opposed to σ_a^i, which was the *total* score of a based on the first i *manipulators* in $\hat{N}$). The X_i are i.i.d. random variables with expectation 0, which take values in $[-\alpha_1, \alpha_1]$. Applying Lemma 7.3.9 to these variables, we obtain:

$$\Pr[\sigma_a - \sigma_b \geq \epsilon n] = \Pr\left[\frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} X_i \geq \mathbb{E}[X_i] + \frac{\epsilon n}{\bar{n}}\right] \leq e^{-2\bar{n}\frac{\left(\frac{\epsilon n}{\bar{n}}\right)^2}{(2\alpha_1)^2}} = e^{-\epsilon' \frac{n^2}{\bar{n}}},$$

where ϵ' is some constant. The result follows from the fact that m is constant and our assumption regarding the relation between n and $\bar{n}$. □

7.4 Related Work

It is possible to identify two main approaches among the preciously few papers on frequency of manipulation: an algorithmic approach and a descriptive approach. In this section we describe at length (at least compared to Section 1.1.1) three papers concerned with the algorithmic approach, whereas in Chapter 8 we discuss the descriptive-oriented papers. Note that the papers we shall discuss were published after the (conference version of the) article on which this chapter is based [119].

Zuckerman et al. [155] extended the results presented in this chapter. Indeed, they did not continue the investigation of Junta distributions, but rather built upon the basic mathematical idea of characterizing the error windows of algorithms for CWM, that is understanding the instances on which an algorithm errs. Note, for example, that this is in fact what our Lemma 7.3.13 is about. Zuckerman et al. did not assume a constant number of alternatives, and managed to achieve rather precise bounds on the error windows of algorithms for CWM under Borda, Maximin, and Plurality with Runoff. With respect to Borda, Zuckerman et al. investigated the greedy algorithm given in this chapter as Algorithm 7.3.1. Their algorithm for Maximin is an immediate generalization of the greedy algorithm, but the algorithm for Plurality with Runoff is based on completely different ideas. Their bounds on error windows also translate to approximation results when it comes to the unweighted coalitional manipulation problem.

Erdélyi et al. [46] discussed the notion of Junta distributions at length. They showed that the idea of Junta distributions, when applied to the SAT problem, is not sufficient to classify hard-to-decide distributions. In more detail, they demonstrated that SAT has a Junta distribution and a heuristic polynomial time algorithm with respect to this distribution. On the other hand, SAT is believed to be hard under many typical distributions. However, the distribution defined by Erdélyi et al. only satisfies the first three properties of a Junta distribution, as the last two are specific to coalitional manipulation. Therefore, their work is somewhat inconclusive when it comes to the application of Junta distributions to CWM.

Finally, an interesting approach to frequency of manipulation was presented by Conitzer and Sandholm [33]. They noticed that an election instance can be manipulated efficiently if it satisfies two properties: weak monotonicity—a property that is always satisfied by many prominent SCFs—and another, more arguable property: the manipulators must be able to make one of *exactly* two alternatives win the election. Conitzer and Sandholm empirically showed that the second property holds with high probability in different standard SCFs. This empirical validation was carried out only with respect to small coalitions of agents and skewed distributions over election instances.

7.5 Discussion

The basic idea behind this chapter is that Junta distributions are in some way representative of other typical distributions over instances of CWM, and therefore the existence of a heuristic polynomial time algorithm with respect to a Junta distribution over instances of CWM suggests that the problem is frequently easy under typical distributions. While this conclusion is certainly arguable at this point, we feel that the definitions presented in this chapter do give a compelling mathematical framework in which frequency of manipulation can be studied.

Some evidence to the validity of our approach, which can perhaps be thought of as a "sanity check", was presented in Section 7.3.3. This section deserves a short discussion. Why is the uniform

distribution interesting? For example, if there are few manipulators relative to nonmanipulators, it is intuitively clear that the manipulators would rarely be able to affect the outcome of the election. Hence, the trivial algorithm, that only looks at the ballots cast by $\bar{N}$ and answers "yes" if and only if p wins based on these ballots, is a heuristic polynomial time algorithm with respect to the uniform distribution. However, the nontrivial aspect of Section 7.3.3 is that the greedy algorithm, that is not tailor made for the uniform distribution (in contrast to the above trivial algorithm), still succeeds with high probability.

The next chapter, Chapter 8, generalizes and proves the above statement that the trivial algorithm succeeds with high probability. The results in Chapter 8 imply that this is true for a very large range of typical distributions that, naturally, does not include our Junta distribution. We feel that this constitutes more evidence to suggest that Junta distributions are especially hard to decide.

Chapter 8

The Fraction of Manipulators

8.1 Introduction

In Chapter 7 we presented and discussed an algorithmic approach to the question of frequency of coalitional manipulation, via the concepts of Junta distributions and heuristic polynomial time algorithms. In particular, we have characterized the behavior of the greedy algorithm with respect to a Junta distribution. However, it might seem more natural to investigate the performance of algorithms with respect to specific typical distributions, such as the uniform distributions.

In this chapter, we take a descriptive approach to frequency of manipulation. We show that, under some assumptions, deciding the coalitional manipulation problem can trivially be accomplished with high probability of success, simply by comparing the number of manipulators and the number of nonmanipulators ("truthful agents"). Indeed, if the fraction of manipulators (out of the total number of agents) is small, the manipulators can rarely influence the outcome of the election at all. On the other hand, if the fraction is large, the manipulators can often change the outcome. As in Chapter 7, the results in this chapter only hold for scoring functions (see Section 2.2.1) and a constant number of alternatives m.

More precisely, an instance of CWM (see Definition 7.1.1) is a *closed instance* if the manipulators cannot affect the outcome of the election. Formally:

Definition 8.1.1. An instance of CWM under f is a *closed* instance if there exists $a \in A$ such that for every $R^{\hat{N}} \in \mathcal{L}^{\hat{N}}$, $f(R^N) = a$. An instance that is not a closed instance is called an *open* instance.

Naturally, knowing whether an instance is closed goes a long way towards deciding CWM. For example, a closed instance is a "yes" instance if and only if the distinguished alternative in Definition 8.1.1 is the preferred alternative p.

Since we will mostly be interested in whether instances of CWM are open or closed, the only parameters of the problem that are not given are the votes of the nonmanipulators $R^{\bar{N}}$, and the weights of the agents. However, as in [33], we prove sufficient conditions for openness/closedness that depend only on the total weight of the manipulators; the individual weights of the manipulators are of no importance. Therefore, weights are a nonissue, and it is sufficient to consider distributions over the possible votes of the nonmanipulators in $\bar{N}$.

8.2 Fraction of Manipulators is Small

As in Chapter 7, we denote $n = \hat{n} = |\hat{N}|$, and $\bar{n} = |\bar{N}|$. In this section we shall demonstrate that when the fraction of manipulators is small, that is $n = o(\sqrt{\bar{n}})$, then usually instances of CWM are closed. This result holds for all scoring functions, and requires only weak assumptions on the distribution of votes.

Given an instance of CWM under a scoring function f, consider the scores of alternatives based only on the votes of the nonmanipulators $\bar{N}$. If there is an alternative whose score is higher than the score of others by more than $\alpha_1 n$, then the instance is surely closed: even if all manipulators ranked this alternative last and another alternative first, the difference in scores would decrease by at most $\alpha_1 n$, which is not enough to close the gap. Further, denote by S_a^i the score given to alternative a by $i \in \bar{N}$. Now, the total score, based on the votes of $\bar{N}$, of alternative $a \in A$ is given by $\sum_{i \in \bar{N}} S_a^i$. We have established the following sufficient condition for closedness:

Lemma 8.2.1. *Consider an instance of CWM under a scoring function with parameters α. Let S_a^i be the score given to alternative $a \in A$ by the agent $i \in \bar{N}$. If there exists an alternative $a \in A$ such that for all $b \in A \setminus \{a\}$, $\sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i > \alpha_1 n$, then the instance is closed.*

Let ρ^i be a distribution over the ballot R^i of agent $i \in \bar{N}$; denote the joint distribution over the votes of the manipulators by $\rho^{\bar{N}} = \prod_{i \in \bar{N}} \rho^i$. ρ^i induces a random variable S_a^i, which determines the points agent i awards to alternative $a \in A$.

Example 8.2.2. Let the f be Borda, and $m = 3$: each agent awards 2 points to its first choice, 1 point to its second choice, and 0 points to its last. If ρ^i is the uniform distribution, then for all $a \in A$, S_a^i is 2 with probability $\frac{1}{3}$, 1 with probability $\frac{1}{3}$, and 0 with probability $\frac{1}{3}$.

We are now ready to present our result.

Theorem 8.2.3. *Let f be a scoring function with parameters α, and assume that the number of manipulators and nonmanipulators satisfies:*

- $n = o(\sqrt{\bar{n}})$.

Let ρ^i be the distribution of agent $i \in \bar{N}$ over the possible votes with $m = \mathcal{O}(1)$ alternatives, and denote $\rho^{\bar{N}} = \prod_{i \in \bar{N}} \rho^i$. Let S_a^i, for each $i \in \bar{N}$ and $a \in A$, be random variables, induced by the ρ^i, which determine the score of alternative a from agent i. Assume that the distributions over votes satisfy:

- ***(d1)*** *There exists a constant $\beta > 0$ such that for all $i \in \bar{N}$ and $a, b \in A$, $\beta < Var[S_a^i - S_b^i]$.*
- ***(d2)*** *The ρ^i are independently distributed.*

Then the probability that an instance is closed converges to 1 as the number of agents grows.

The proof relies heavily on the central limit theorem. For our purposes, this theorem implies that the probability that a sum of random variables obtain values in a very small segment is very small, as long as the variance of the random variables is nonzero.

Theorem 8.2.4 (Central Limit Theorem)**.** *[54] Let*

$$X^1, \ldots, X^t, \ldots$$

be a sequence of independent discrete random variables. For each i, denote the mean and variance of X^i by μ^i and σ^i, respectively, and assume that $\sum_{i=1}^t \sigma^i \overset{t\to\infty}{\longrightarrow} \infty$, and that $|X^i| \le K$ for some constant K and all i. Then for $\beta < \gamma$:

$$\Pr\left[\beta < \frac{\sum_{i=1}^t X^i - \sum_{i=1}^t \mu^i}{\sqrt{\sum_{i=1}^t \sigma^i}} < \gamma\right] \overset{t\to\infty}{\longrightarrow} \frac{1}{\sqrt{2\pi}} \int_\beta^\gamma e^{-\frac{x^2}{2}} dx \ .$$

Proof of Theorem 8.2.3. By Lemma 8.2.1 we have:

$$\begin{aligned}
\Pr_{\rho^{\bar{N}}}[\text{instance is closed}] &\ge \Pr_{\rho^{\bar{N}}}\left[\exists a \in A \ s.t. \ \forall b \in A \setminus \{a\}, \ \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i > \alpha_1 n\right] \\
&\ge \Pr_{\rho^{\bar{N}}}\left[\forall a \in A, b \in A \setminus \{a\}, \left|\sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i\right| > \alpha_1 n\right] \\
&= 1 - \Pr_{\rho^{\bar{N}}}\left[\exists a \in A, b \in A \setminus \{a\} \ s.t. \ 0 \le \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \le \alpha_1 n\right] \ .
\end{aligned}$$

Now, by the union bound, we have that

$$\Pr_{\rho^{\bar{N}}}\left[\exists a \in A, b \in A \setminus \{a\} \ s.t. \ 0 \le \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \le \alpha_1 n\right] \le \sum_{a\neq b} \Pr_{\rho^{\bar{N}}}\left[0 \le \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \le \alpha_1 n\right] \ . \tag{8.1}$$

Fix two alternatives $a \in A$, $b \in A\setminus\{a\}$, and denote $X^i = S_a^i - S_b^i$. Let $\mu^i = \mathbb{E}[X^i]$, $\sigma^i = \mathrm{Var}[X^i]$. Notice that $\sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i = \sum_{i\in\bar{N}} X^i$. In addition, observe that by assumption (d1) $\beta < \sigma^i$, and thus $\sum_{i\in\bar{N}} \sigma^i \overset{\bar{n}\to\infty}{\longrightarrow} \infty$. Finally, for all $i \in \bar{N}$, $|X^i| \le \alpha_1$. Therefore, we may apply Theorem 8.2.4 to the variables X^i.

$$\begin{aligned}
&\Pr_{\rho^{\bar{N}}}\left[0 \le \sum_{i\in\bar{N}} X^i \le \alpha_1 n\right] = \Pr_{\rho^{\bar{N}}}\left[\frac{-\sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}} \le \frac{\sum_{i\in\bar{N}} X^i - \sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}} \le \frac{\alpha_1 n - \sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}}\right] \\
&\overset{N\to\infty}{\longrightarrow} \frac{1}{\sqrt{2\pi}} \int_{\frac{-\sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}}}^{\frac{\alpha_1 n - \sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}}} e^{-\frac{x^2}{2}} dx \le \int_{\frac{-\sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}}}^{\frac{\alpha_1 n - \sum_{i\in\bar{N}} \mu^i}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}}} 1 \ dx = \frac{\alpha_1 n}{\sqrt{\sum_{i\in\bar{N}} \sigma^i}} \le \frac{\alpha_1 n}{\sqrt{\beta \bar{n}}} = \mathcal{O}\left(\frac{n}{\sqrt{\bar{n}}}\right).
\end{aligned}$$

Plugging this result into (8.1), we have that

$$\Pr_{\rho^{\bar{N}}}\left[\exists a \in A, b \in A \setminus \{a\} \ s.t. \ 0 \le \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \le \alpha_1 n\right] \le m(m-1) \cdot \mathcal{O}\left(\frac{n}{\sqrt{\bar{n}}}\right) = \mathcal{O}\left(\frac{n}{\sqrt{\bar{n}}}\right),$$

where the second transition follows from the fact that m is constant. Rolling back, we have that

$$\Pr_{\rho^{\bar{N}}}[\text{instance is closed}] \ge 1 - \mathcal{O}\left(\frac{n}{\sqrt{\bar{n}}}\right).$$

Under the assumption that $n = o(\sqrt{\bar{n}})$, this expression converges to 1 as the number of agents grows. □

8.3 Fraction of Manipulators is Large

In this subsection, we tackle a setting where the number of manipulators is large, i.e., $n = \omega(\sqrt{\bar{n}})$, but not exceedingly so, i.e., $n = o(\bar{n})$. The mathematical techniques we use here differ from the ones applied in Section 8.2.

As before, we characterize instances of the manipulation problem in scoring functions. Crucially, in the current setting, the manipulators may often have enough power to sway the outcome of the election. Therefore, we require a sufficient condition for the openness of a manipulation instance.

Lemma 8.3.1. *Consider an instance of the coalitional manipulation problem in a scoring function with parameters α, and assume $n \geq m$. Let S_a^i be the score given to alternative a by the agent $i \in \bar{N}$. Let $A' \subseteq A$ such that for any two alternatives $a, b \in A'$ it holds that $\sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i < \frac{\alpha_1-\alpha_m}{2m} \cdot n$, and for any $a \in A'$ and $b \notin A'$, $\sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \geq 0$. Then the manipulators can make any alternative in A' win.*

Proof. Let $A = \{a_0, \ldots, a_{m-2}, p\}$, and assume w.l.o.g. that the manipulators wish to make alternative $p \in A'$ win; the manipulators $\bar{N}$ vote as follows. The i'th manipulator, $i = 1, \ldots, n$, ranks p first, $a_{i \bmod (m-1)}$ last, and the other alternatives in some arbitrary order. Each alternative other than p is ranked last by at least $\left\lfloor \frac{n}{m-1} \right\rfloor$ manipulators, and the rest of the manipulators award it at most α_1 points. Therefore, the difference in the points awarded by the manipulators to p and any other alternative is at least $\left\lfloor \frac{n}{m-1} \right\rfloor \cdot (\alpha_1 - \alpha_m) \geq \frac{\alpha_1-\alpha_m}{2m} \cdot n$, where the inequality holds whenever $n \geq m$. □

Our theorems regarding the current setting are weaker than the ones in Section 8.2, in the sense that the votes of the agents are (independent and) identically distributed. The following theorem differentiates two cases: if there are at least two alternatives whose expected score is at least as large as that of any other alternative, then the instance is open; otherwise, the instance is closed. Intuitively, whenever the first case holds, one of the alternatives with a large expected score will surely win, but the manipulators are powerful enough to decide between them. However, if there is an alternative whose expected score is greater than that of any other, even a large fraction of manipulators cannot prevent this alternative from winning.

Theorem 8.3.2. *Let f be a scoring function with parameters α, and assume that the number of manipulators and nonmanipulators satisfies:*

- *$n = \omega(\sqrt{\bar{n}})$ and $n = o(\bar{n})$.*

Let ρ^i be the distribution of agent $i \in \bar{N}$ over the possible votes with $m = \mathcal{O}(1)$ alternatives, and denote $\rho^{\bar{N}} = \prod_{i\in\bar{N}} \rho^i$. Let S_a^i, for each $i \in \bar{N}$ and $a \in A$, be random variables, induced by the ρ^i, which determine the score of alternative a from agent i. Assume that the distributions over votes satisfy:

- ***(d2)*** *The ρ^i are independently distributed*
- ***(d3)*** *The ρ^i are identically distributed.*

Let $A' = \{a \in A : \ \forall b \in A \setminus \{a\}, \ \mathbb{E}[S_a^1] \geq \mathbb{E}[S_b^1]\}$ be the subset of alternatives with maximum expected score.

1. *If $|A'| \geq 2$, then the probability of drawing an open instance converges to 1 as the number of agents grows.*

2. *If $|A'| = 1$ then the probability of drawing a closed instance converges to 1 as the number of agents grows.*

Proof. For part 1, assume $|A'| \geq 2$. Using Lemma 8.3.1 and denoting $\delta = \frac{\alpha_1 - \alpha_m}{2m}$, and also applying Lemma 8.2.1, we obtain:

$$\begin{aligned}
&\Pr_{\rho^{\bar{N}}}[\text{Instance is open}] \geq \Pr_{\rho^{\bar{N}}}[a \in A \text{ can be made to win iff } a \in A'] \\
&\quad \geq \Pr_{\rho^{\bar{N}}}\left[\left(\forall a, b \in A',\ \sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i < \delta n\right) \wedge \left(\forall a \in A', b \in A \setminus A',\ \sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i > \alpha_1 n\right)\right] \\
&\quad = 1 - \Pr_{\rho^{\bar{N}}}\left[\left(\exists a, b \in A' \ s.t.\ \sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i \geq \delta n\right)\right. \\
&\qquad\qquad \left. \vee \left(\exists a \in A', b \in A \setminus A' \ s.t.\ \sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i \leq \alpha_1 n\right)\right]
\end{aligned} \tag{8.2}$$

Now, it holds that:

$$\begin{aligned}
&\Pr_{\rho^{\bar{N}}}\left[\exists a, b \in A' \ s.t.\ \sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i \geq \delta n\right] \leq \sum_{a,b \in A'} \Pr_{\rho^{\bar{N}}}\left[\sum_{i \in \bar{N}} S_a^i - \sum_{i \in \bar{N}} S_b^i \geq \delta n\right] \\
&\quad = \sum_{a,b \in A'} \Pr_{\rho^{\bar{N}}}\left[\sum_{i \in \bar{N}} (S_a^i - S_b^i) \geq \mathbb{E}\left[\sum_{i \in \bar{N}} (S_a^i - S_b^i)\right] + \delta n\right] \\
&\quad \leq |A'| \cdot (|A'| - 1) \cdot e^{\frac{-2\bar{n}\left(\frac{\delta n}{\bar{n}}\right)^2}{(2\alpha_1)^2}} \leq m(m-1)e^{-\delta_1 \frac{n^2}{\bar{n}}} = \mathcal{O}\left(e^{-\delta_1 \frac{n^2}{\bar{n}}}\right)
\end{aligned} \tag{8.3}$$

for some constant $\delta_1 > 0$. The first transition follows from the union bound, the second from the fact that all alternatives in A' have maximum expected score and the linearity of expectation, and the third from Chernoff's bounds (Lemma 7.3.9, where we use the fact that the difference between the scores given to two alternatives by an agent is in the range $[-\alpha_1, \alpha_1]$, and that the S_a^i are i.i.d. for a fixed $a \in A$ if the ρ^i are i.i.d.).

Further, we have that:

$$\begin{aligned}
&\Pr_{\rho^{\bar{N}}}\left[\exists a \in A', b \in A \setminus A' \ s.t. \ \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \leq \alpha_1 n\right] \leq \sum_{a\in A', b\in A\setminus A'} \Pr_{\rho^{\bar{N}}}\left[\sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \leq \alpha_1 n\right] \\
&= \sum_{a\in A', b\in A\setminus A'} \Pr_{\rho^{\bar{N}}}\left[\sum_{i\in\bar{N}}(S_a^i - S_b^i) \leq \mathbb{E}\left[\sum_{i\in\bar{N}}(S_a^i - S_b^i)\right] - \left(\mathbb{E}\left[\sum_{i\in\bar{N}}(S_a^i - S_b^i)\right] - \alpha_1 n\right)\right] \\
&\leq |A'| \cdot (m - |A'|) \cdot e^{\frac{-2\bar{n}\left(\frac{\delta'\bar{n} - \alpha_1 n}{\bar{n}}\right)^2}{(2\alpha_1)^2}} = \mathcal{O}\left(e^{-\delta_2 \bar{n}}\right) \ .
\end{aligned} \tag{8.4}$$

The first transition follows from the union bound. The third transition is entailed by Chernoff's bounds (Lemma 7.3.9), where δ' is a constant such that $\mathbb{E}[S_a^i - S_b^i] \geq \delta'$ for all $a \in A', b \in A \setminus A'$.[1] The last transition follows from the assumption that $n = o(\bar{n})$; $\delta_2 > 0$ is a constant.

Combining (8.2), (8.3), and (8.4), and applying the union bound, we get:

$$\Pr_{\rho^{\bar{N}}}[\text{The instance is open}] \geq 1 - \left(\mathcal{O}\left(e^{-\delta_1 \frac{n^2}{\bar{n}}}\right) + \mathcal{O}\left(e^{-\delta_2 \bar{n}}\right)\right) \ .$$

When $n = \omega(\sqrt{\bar{n}})$, this expression converges to 1 as the number of agents grows.

For part 2, assume $A' = \{a\}$. By Lemma 8.3.1, we have:

$$\begin{aligned}
\Pr_{\rho^{\bar{N}}}[\text{instance is closed}] &\geq \Pr_{\rho^{\bar{N}}}\left[\forall b \in A \setminus \{a\}, \ \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i > \alpha_1 n\right] \\
&= 1 - \Pr_{\rho^{\bar{N}}}\left[\exists b \in A \setminus \{a\} \ s.t. \ \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \leq \alpha_1 n\right] \ .
\end{aligned} \tag{8.5}$$

Similarly to (8.4), it holds that

$$\Pr_{\rho^{\bar{N}}}\left[\exists b \in A \setminus \{a\} \ s.t. \ \sum_{i\in\bar{N}} S_a^i - \sum_{i\in\bar{N}} S_b^i \leq \alpha_1 n\right] \leq \mathcal{O}(e^{-\delta \bar{n}}).$$

Plugging this into (8.5) gives the desired result.

□

The next corollary establishes a useful connection between the proof of Theorem 8.3.2 and the decision of the coalitional manipulation problem.

Corollary 8.3.3. *Under the conditions of Theorem 8.3.2, if A' is the set of alternatives with maximum expected score, then with probability that converges to 1 it holds that any alternative from A' can be made to win, and no other alternative can be made to win.*

Algorithm 8.4.1 Deciding the coalitional manipulation problem in scoring functions via the fraction of manipulators. The input is an instance drawn according to a distribution over the votes of the nonmanipulators; p is the preferred alternative of the manipulators.

if $n = o(\sqrt{\bar{n}})$ **then** ▷ Theorem 8.2.3
  choose arbitrary vote $R^{\hat{N}}$ for manipulators
  $a \leftarrow f(R^N)$ ▷ a is the winner under the arbitrary vote
  if $p = a$ **then**
    return true
  else
    return false
  end if
else if $n = \omega(\sqrt{\bar{n}})$ and $n = o(\bar{n})$ **then** ▷ Theorem 8.3.2
  if p has maximum expected score **then**
    return true
  else
    return false
  end if
else ▷ $n = \Theta(\sqrt{\bar{n}})$ or $n = \Omega(\bar{n})$
  return ?
end if

8.4 Algorithmic Implications

Consider Algorithm 8.4.1, which instantly decides instances of the coalitional manipulation problem, drawn according to some distribution, on the basis of the ratio between the number of manipulators n and nonmanipulators $\bar{n}$. Theorems 8.2.3 and 8.3.2 directly imply that for any distribution that satisfies assumptions (d1), (d2), and (d3), Algorithm 8.4.1 is almost never wrong when the number of agents is large. Indeed, when $n = o(\sqrt{\bar{n}})$, Theorem 8.2.3 asserts that instances are almost always closed—and therefore p can be made to win iff p wins for any arbitrary vote of the manipulators. In case $n = \omega(\sqrt{\bar{n}})$, Corollary 8.3.3 states that it is usually true that the manipulators can only make alternatives with maximal expected score win the election.

But how restrictive are the assumptions (d1), (d2), and (d3)? Assumption (d1) requires that there exist a constant $\beta > 0$ such that for all $i \in \hat{N}$ and distinct alternatives $a, b \in A$, $\beta < \mathrm{Var}[S_a^i - S_b^i]$. This is certainly a condition that seems very reasonable: the demand is that according to the distribution of each agent, there are no two alternatives that *always* have the same difference in scores. That is, we simply require a seemingly minimal element of randomness in the votes. Granted, requiring that the votes of the nonmanipulators be distributed i.i.d.—the union of assumptions (d2) and (d3)—is a much stricter assumption. Nevertheless, we argue below that interesting distributions satisfy all three assumptions.

First, it is obvious that the i.i.d. uniform distribution satisfies all three assumptions. Specifically, the probability of an agent casting a specific ballot is $1/m!$, and this holds for every possible ranking. This flavor of the uniform distribution is sometimes known as the *Impartial Culture Assumption* (see, e.g., Slinko [141]).

[1]It is safe to state that such a constant δ' exists, as we assumed that $\mathbb{E}[S_a^i - S_b^i] > 0$, and D is (implicitly) a distribution that is dependent only on the number of alternatives m, and not on the number of agents.

As a second example, we shall consider the family of distributions that Conitzer and Sandholm used to obtain empirical evidence regarding the nonexistence of SCFs that are frequently hard to manipulate [33]; these distributions are due to the Marquis de Condorcet himself. The starting point is that there is a "correct" ranking of alternatives Q, and agents disagree with this ranking over pairs of alternatives with probability q. More formally, the probability of an agent casting a vote R is proportional to

$$q^{\Delta(R,Q)}(1-q)^{m(m-1)/2-\Delta(R,Q)}, \tag{8.6}$$

where $\Delta(R,Q)$ is the number of pairs of alternatives on whose relative ranking R and Q agree. The parameter q can take values in $[1/2,1]$: if $q=1$ then the agents always agree with the correct ranking, and if $q=1/2$ then agents vote randomly. Of course, the expression in (8.6) above has to be normalized in order to obtain a probability distribution.

Proposition 8.4.1. *Let $m=\mathcal{O}(1)$. Condorcet's distribution with any $0.5\leq q<1$ satisfies (d1), (d2), and (d3).*

Proof. By definition, the distribution satisfies (d2) and (d3), so it is sufficient to prove that (d1) is satisfied. Let $i\in\bar{N}$, $a,b\in C$, and let $\beta'>0$ be a constant such that dividing the expression in (8.6) by β' yields a probability distribution. Let Q be the "correct" ranking of alternatives, and consider the restriction of Q to all alternatives other than a,b. Now, let R_1 be the expansion of this ranking such that a is ranked first and b last, and let R_2 be the expansion such that b is ranked first and a is ranked last. Under R_1 it holds that $S_a^i-S_b^i=\alpha_1-\alpha_m$, while under R_2 it holds that $S_a^i-S_b^i=\alpha_m-\alpha_1$. Therefore, with respect to at least one of R_1 and R_2, it is true that $|(S_a^i-S_b^i)-\mathbb{E}[S_a^i-S_b^i]|\geq\alpha_1-\alpha_m$—w.l.o.g. with respect to R_1. By the construction of R_1, this ranking can differ from Q only on pairs of alternatives which include a or b, i.e., $\Delta(R_1,Q)\leq 2(m-1)$. Therefore, $\Pr[i\text{'s ballot } R^i \text{ is } R_1]\geq\frac{q^{2(m-1)}(1-q)^{(\frac{m}{2}-2)(m-1)}}{\beta'}$; denote this (constant) expression by β. To conclude, we have obtained:

$$\mathrm{Var}[S_a^i-S_b^i]=\mathbb{E}[((S_a^i-S_b^i)-\mathbb{E}[S_a^i-S_b^i])^2]\geq\Pr[R^i=R_1]\cdot(\alpha_1-\alpha_m)^2\geq\beta(\alpha_1-\alpha_m)^2.$$

□

Finally (and crucially), the Junta distribution presented in Chapter 7 satisfies (d2) and (d3) but not (d1): the votes of the nonmanipulators are distributed in an interval which is proportional to the number of manipulators, and thus the variance can be very small in terms of the number of nonmanipulators $\bar{n}$. This of course makes the distribution harder to manipulate, otherwise Theorem 8.2.3 could have been used to decide instances distributed with respect to our Junta distribution.

8.5 Related Work

Recall that in Section 7.4 we have discussed a number of works on frequency of manipulation in elections. Those works were algorithmic in nature, and therefore more related to the results presented in Chapter 7. In this section we describe some papers regarding frequency of manipulation that we categorize as *descriptive* rather than algorithmic.

Previous work in economics has independently recognized that when the fraction of manipulators is small, manipulation is rarely possible [5, 141]. However, these papers consider only variations

on the uniform distribution over possible ballots; this is plausible from the point of view of the economist, but in computer science we are interested in the behavior of the problem under a range of typical distributions; indeed, finding an SCF that is frequently hard to manipulate even under one typical distribution would be an accomplishment. Additionally, unlike the abovementioned work, we present our results and their implications from a computational point of view. In particular, the formulation of the manipulation problem that we consider is the one generally accepted in computer science.

A very interesting extension of our work was presented by Xia and Conitzer [147]. They define a class of SCFs that they call *Generalized Scoring Functions*. A generalized scoring function transforms the vote of each agent to a vector of scores of length k, for some fixed k, then sums the vectors generated by the agents coordinate-wise, and applies a function on the final vector to determine the winner of the election. This class of SCFs captures almost all conceivable anonymous SCFs (i.e. those SCFs that disregard the identity of the agents), and indeed includes all the prominent SCFs mentioned in Chapter 2.2. The authors demonstrate that our results, namely Theorems 8.2.3 and 8.3.2, hold for all generalized scoring functions rather than just scoring functions. The assumptions that they use are basically identical to ours. Hence, their results hold under the same class of distributions, but for a significantly larger variety of SCFs.

Friedgut et al. [57] proposed another fascinating direction, albeit in the context of manipulation by a single agent rather than a coalition. They suggested that the probability, over choices of random preference profiles, that a manipulator would be able to improve the outcome of the election by a random ballot is non-negligible. Interestingly, the probability of success depends on the distance of the given SCF from dictatorship, that is, the fraction of preference profiles on which the function must be redefined in order to make it a dictatorship. All prominent SCFs are far from being a dictatorship, implying that the probability of success is significant. Unfortunately, the results of Friedgut et al. only hold for $m = 3$. Xia and Conitzer [148] extended the results of Friedgut et al. to any constant number of alternatives, but added additional, rather restrictive, assumptions with respect to the SCF.

8.6 Discussion

Our results are satisfied by a wide spectrum of distributions over votes. Still, there remain gray areas, even when considering distributions that satisfy all three conditions: Theorems 8.2.3 and 8.3.2 do not apply to situations where $n = \Theta(\sqrt{\bar{n}})$ or $n = \Omega(\bar{n})$. The latter case, where $n = \Omega(\bar{n})$, does not seem very interesting: it is quite clear that when the number of manipulators is that large, the manipulators can usually determine the outcome of the election. However, the former case, where $n = \Theta(\sqrt{\bar{n}})$, persists as a wide-open question (in our work as well as in the extension by Xia and Conitzer [147]). In fact, if a distribution which cannot be frequently manipulated were to exist (especially in the context of scoring functions), our belief is that the distribution would be over instances that satisfy $n = \Theta(\sqrt{\bar{n}})$.

To conclude, there are two ways to interpret our results. A positive interpretation would be that a distribution that is frequently hard to manipulate exists, and the results may simply help focus the search for such a distribution. Interpreted negatively, these results strengthen the case against the existence of SCFs and distributions that are frequently hard to manipulate. Indeed, our results, and even more so the subsequent results of Xia and Conitzer, imply that the manipulation problem under many SCFs can usually be trivially decided, with respect to a wide range of distributions.

Chapter 9

Conclusions

We shall use this chapter to succinctly lay out our view of the future of the field of Computational Voting Theory. A lot of the work in Social Choice Theory has concentrated on impossibility results. Similarly, a lot of the previous work in Computational Voting Theory has focused on computational hardness results. These results can sometimes be interpreted positively, e.g., in the context of hardness of manipulation, but are usually negative, e.g., in the context of winner determination, computing possible and necessary winners, etc.

We believe, on the other hand, that the focus should be using computer science techniques and notions to obtain novel positive results in Voting Theory. This understanding is perhaps what initiated the research on using computational hardness to preclude manipulation [8], but it is now becoming quite clear that this agenda cannot necessarily be justified in practice, as discussed in Chapter 7. In the following we outline two promising new agendas that are related to the work presented in this thesis and exemplify the above discussion, that is, using computer science paradigms (in particular, approximation) to obtain positive results in voting theory.

Approximation in mechanism design without money. We have noted in Section 6.7 that one of the main contributions of our work on strategyproof learning is the notion of approximation in mechanism design without money. In settings where the preferences of the agents are restricted, the Gibbard-Satterthwaite Theorem [60, 135] does not hold. Hence, it is possible to achieve strategyproof mechanisms.

For instance, imagine the paradigmatic example of single-peaked preferences. The agents have ideal points on the real line, which represent, e.g., their locations or where they live. The mechanism must choose a point, e.g., a location for a grocery store. The cost of each agent is its distance from the chosen location. The preferences of the agents are completely encoded by their ideal point, so it is sufficient to report these points. Notice that the mechanism that always chooses the leftmost reported point is strategyproof. However, it does not make a lot of sense in terms of the social welfare, that is, the total cost of the agents. On the other hand, choosing the median point is also strategyproof, and it can be easily verified that this solution minimizes the total cost.

Let us now complicate this situation. We now wish to select two points (e.g., select two locations for facilities). The utility of an agent is its distance to the closest facility. Schummer and Vohra [136] have asked whether there are strategyproof mechanisms in this setting. One strategyproof solution is placing the facilities at the leftmost reported point and the rightmost reported point, but this is very far from minimizing the total cost. On the other hand, it can be verified that choosing the

locations that minimize the total cost is not strategyproof.

Now we are exactly on the boundary between social choice and computer science: social choice theory does not have tools to deal with this situation. We suggest that a computer scientist should naturally ask: given the above setting (with two facility locations to be selected), is there a mechanism that is strategyproof and *approximates* the minimum total cost? Similar questions can be asked when the goal is to minimize the maximum cost, rather than the total cost. In this case, even placing one facility optimally is not strategyproof. In very recent and ongoing work with Moshe Tennenholtz, we have obtained some preliminary answers to these questions.

More generally, the same agenda can be applied to many mechanism design settings that are computationally tractable. Such settings were previously disregarded by computer scientists since the VCG mechanism [146, 25, 62] can be applied to obtain strategyproofness while maximizing social welfare. An example is the shortest path network domain briefly considered by Nisan and Ronen in their seminal paper [105]. However, and crucially, the transfer of payments is infeasible in many (most?) multi-agent settings, especially in extremely distributed Internet environments. Once again, it is possible to ask: is there a strategyproof solution without payments that approximates the social welfare, or some other target function? In other words, we can sacrifice optimality in terms of our optimization goal in order to gain strategyproofness.

Approximation algorithms as SCFs. Many previous works demonstrate that it is hard to compute the outcome of the election under various SCFs (e.g., [66, 132, 127]). Nevertheless, many of the hard-to-compute score-based SCFs can be approximated. Chapter 3 presented a randomized rounding approximation algorithm for Dodgson's rule. We noted that more recent work on the problem includes a deterministic approximation algorithm with the same approximation ratio of $\mathcal{O}(\log m)$. These can be considered positive, encouraging results: while it is computationally intractable to elect the alternative closest to being a Condorcet winner, it is possible to elect an alternative that is quite close.

Nevertheless, such results open a window to many interesting issues. If approximation algorithms are to be used as SCFs, they must satisfy at least some of the desirable properties SCFs are expected to satisfy. We mention several important properties below:

- *Anonymity*: The identities of the agents are disregarded.
- *Neutrality*: The identities of the alternatives are disregarded.
- *Monotonicity*: Improving the position of an alternative in a preference profile without changing the order of the other alternatives cannot hurt the improved alternative, that is, if it was a winner in the original profile it would also be a winner under the improvement.
- *Homogeneity*: Duplicating the electorate does not change the outcome of the election.[1]

Designing algorithms that satisfy these properties while achieving a good approximation ratio for the score under a hard-to-compute SCF seems to be a conceptually novel and nontrivial agenda.

[1] Anonymity must be assumed in order for this definition to be sound.

Appendix

Appendix A

Omitted Proofs and Results for Chapter 4

A.1 Proof of Theorem 4.3.3

Proof. Reformulating the minimax principle for voting trees, an upper bound on the worst-case performance of the best randomized tree on a set A of alternatives is given by the performance of the best deterministic tree with respect to some probability distribution over tournaments on A.

As in the proof of Theorem 4.3.2, we assume for ease of exposition that $|A| = m = 3k + 1$ for some odd k, and define a tournament T as a cycle of three regular components C_1, C_2, and C_3, each of size k. Further define three new tournaments T_1, T_2, and T_3 such that for $r = 1, 2, 3$, the restrictions of T and T_r to $B \subseteq A$ are identical if $|B \cap C_r| \leq 1$, and the restriction of T_r to C_r is transitive. Let Γ be any deterministic tree on A. Combining both statements of Lemma 4.3.1, there exists $i \in \{1, 2, 3\}$ such that for $r = 1, 2, 3$, $\Gamma(T_r) \in C_i$. In particular, Γ selects an alternative with score at most $3k/2 - 1/2$ for two of the three tournaments T_r. Now consider a tournament T drawn uniformly from $\{T_1, T_2, T_3\}$. By the above,

$$\mathbb{E}_{\Gamma\sim\Delta}[s_{\Gamma(T)}] \leq (2(3k/2 - 1/2) + (2k - 1))/3 = 5k/3 - 2/3 \qquad \text{and} \qquad \max_{i\in A} s_i = 2k - 1,$$

and thus

$$\frac{\mathbb{E}_{\Gamma\sim\Delta}[s_{\Gamma(T)}]}{\max_{i\in A} s_i} \leq \frac{5k-2}{6k-3} \leq \frac{5(k-1)+3}{6(k-1)} = \frac{5}{6} + \frac{1}{2(k-1)}.$$

In particular, this ratio tends to $5/6$ as k tends to infinity. □

A.2 Proof of Theorem 4.4.8

We shall require two lemmata. The first one is a "geometric" version of the Cauchy-Schwarz inequality. The second one is a well-known result about the sequence of degrees of a tournament, which we state without proof.

Lemma A.2.1. *Let* $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{R}^m$, $\mathbf{b} = (b_1, \ldots, b_m) \in \mathbb{R}^m$. *Then,*

$$\sum_{i=1}^{m} \left(\frac{a_i}{\|a\|} - \frac{b_i}{\|b\|} \right)^2 = \epsilon \qquad \textit{if and only if} \qquad \sum_{i=1}^{m} a_i b_i = (1 - \frac{\epsilon}{2})\|a\| \cdot \|b\|.$$

Proof.

$$\sum_{i=1}^{m}\left(\frac{a_i}{\|a\|}-\frac{b_i}{\|b\|}\right)^2=\epsilon \iff \sum_{i=1}^{m}\frac{(a_i)^2}{\|a\|^2}+\sum_{i=1}^{m}\frac{(b_i)^2}{\|b\|^2}-2\sum_{i=1}^{m}\frac{a_i}{\|a\|}\frac{b_i}{\|b\|}=\epsilon$$
$$\iff \sum_{i=1}^{m}\frac{a_i}{\|a\|}\frac{b_i}{\|b\|}=1-\frac{\epsilon}{2}$$
$$\iff \sum_{i=1}^{m}a_ib_i=(1-\frac{\epsilon}{2})\|a\|\cdot\|b\|. \qquad \square$$

$\square$

Lemma A.2.2 (Moon [100])**.** *$s_1 \le s_2 \le \cdots \le s_m$ is the degree sequence of a tournament if and only if for all $k \le m$, $\sum_{i=1}^{k} s_i \ge \binom{k}{2}$.* $\square$

Proof of Theorem 4.4.8. Define $w_i = m - s_i - 1$, $a_i = \sqrt{2w_i+1}$, and $b_i = \sqrt{2w_i+1}\pi_i$. By the assumption that $\sum_i \pi_i s_i = \frac{m-1}{2} + \epsilon m$ and by (4.1) in the proof of Lemma 4.4.5, we have that $\sum_i a_i b_i = (1-2\epsilon)m$. Since $\|a\| = m$ and, by Lemma 4.4.6, $\|b\| = 1$, we have

$$\sum_i a_i b_i = (1-2\epsilon)\|a\|\cdot\|b\|.$$

By Lemma A.2.1,

$$\sum_i\left(\frac{a_i}{\|a\|}-\frac{b_i}{\|b\|}\right)^2 = 4\epsilon.$$

Denoting $\epsilon' = 4\epsilon$,

$$\sum_i\left(\frac{\sqrt{2w_i+1}}{m}-\sqrt{2w_i+1}\cdot\pi_i\right)^2=\epsilon'.$$

By simplifying and rearranging, we get

$$\sum_i(2w_i+1)\left(\pi_i-\frac{1}{m}\right)^2=\epsilon'. \tag{A.1}$$

Now let $\epsilon'' = \sqrt[4]{\epsilon'}$, and

$$B=\left\{i\in A:\ \left|\pi_i-\frac{1}{m}\right|>\frac{\epsilon''}{m}\right\}.$$

We claim that $|B| \le \epsilon'' m$. Assume for contradiction that $|B| > \epsilon'' m$. Then, by Lemma A.2.2,

$$\sum_{i\in B}s_i=\binom{m}{2}-\sum_{i\notin B}s_i\le\binom{m}{2}-\binom{m-|B|}{2},$$

and

$$\sum_{i\in B}w_i\ge|B|(m-1)-\binom{m}{2}+\binom{m-|B|}{2}=\binom{|B|}{2}.$$

We thus have

$$\sum_{i\in B}(2w_i+1)\left(\pi_i-\frac{1}{m}\right)^2 > \frac{\sqrt{\epsilon'}}{m^2}\sum_{i\in B}(2w_i+1) \geq \frac{\sqrt{\epsilon'}}{m^2}\left(2\frac{|B|(|B|-1)}{2}+|B|\right) > \frac{\sqrt{\epsilon'}}{m^2}\cdot\sqrt{\epsilon'}m^2 = \epsilon',$$

contradicting (A.1). The first inequality holds because $|\pi_i - 1/m| > \epsilon''/m$ for all $i \in B$, the last one follows from the assumption that $|B| > \epsilon'' m$.

It now suffices to show that for all $i \notin B$, $\left|s_i - \frac{m}{2}\right| \leq (3\epsilon''/2)m$, i.e., that B contains all alternatives with degree significantly different from $m/2$.

Let $i \in A \setminus B$. Since π is a stationary distribution,

$$(m - s_i - 1)\pi_i = \sum_{j:iTj}\pi_j.$$

At most $\epsilon'' m$ of the alternatives dominated by i can be in B, and thus

$$m - s_i - 1 \geq \frac{(s_i - \epsilon'' m)\left(\frac{1}{m} - \frac{\epsilon''}{m}\right)}{\frac{1}{m}+\frac{\epsilon''}{m}}.$$

It should be noted that this holds even if $s_i - \epsilon'' m < 0$. By rearranging and simplifying,

$$(m - s_i - 1)(1+\epsilon'') \geq (1-\epsilon'')s_i - m\epsilon''(1-\epsilon''),$$

and thus

$$s_i \leq \frac{m}{2} + \epsilon'' m.$$

On the other hand,

$$\sum_{j\notin B}\pi_j \geq (1-\epsilon'')m\cdot\frac{1-\epsilon''}{m},$$

and therefore

$$(m - s_i - 1) \leq \frac{s_i\frac{1+\epsilon''}{m} + \left(1-(1-\epsilon'')m\frac{1-\epsilon''}{m}\right)}{\frac{1-\epsilon''}{m}}.$$

The last implication is true because i dominates at most s_i alternatives outside B, and the overall probability assigned to alternatives in B is at most $1-(1-\epsilon'')m\frac{1-\epsilon''}{m}$. Now,

$$(m - s_i - 1)(1-\epsilon'') \leq s_i(1+\epsilon'') + m(2\epsilon'' - (\epsilon'')^2).$$

Thus, for $m \geq \frac{1}{(\epsilon'')^2}$,

$$s_i \geq \frac{m}{2} - \frac{3}{2}\epsilon'' m. \quad \square$$

$\square$

A.3 Proof of Lemma 4.4.10

Proof. Fix some tournament $T \in \mathcal{T}(A)$, and consider the degrees s_i in T. The minimum expected second order degree of an alternative drawn according to the stationary distribution of $\mathfrak{M}(T)$ is given by the following linear program with variables π_i:

$$\begin{aligned}
\textbf{min} \quad & \sum_{i \in A} \pi_i \left(\sum_{j : iTj} s_j \right) \\
\textbf{s.t.} \quad & \forall i, \ (m - s_i - 1)\pi_i - \sum_{j : iTj} \pi_j = 0, \\
& \sum_{i \in A} \pi_i = 1, \\
& \forall i, \pi_i \geq 0.
\end{aligned}$$

The dual is the following program with variables x_i and y:

$$\begin{aligned}
\textbf{max} \quad & y \\
\textbf{s.t.} \quad & \forall i, \ (m - s_i - 1)x_i - \sum_{j : jTi} x_j + \sum_{j : iTj} s_j \geq y.
\end{aligned}$$

By weak duality, any feasible solution to the dual provides a lower bound on the optimal assignment to the primal. Consider the assignment $x_i = -s_i$ to the dual. The maximum feasible value of y given this assignment is the minimum over the left hand side of the constraints. We claim that for any i, the value of the left hand side is at least $m^2/4 - m/2$. Indeed, for all i,

$$\begin{aligned}
(m - s_i - 1)(-s_i) - \sum_{j : jTi} (-s_j) + \sum_{j : iTj} s_j &= (m - s_i - 1)(-s_i) + \sum_{j \neq i} s_j \\
&= (m - s_i - 1)(-s_i) + \left(\binom{m}{2} - s_i \right) \\
&= m^2/2 - m/2 - s_i(m - s_i) \\
&\geq m^2/4 - m/2.
\end{aligned}$$

□

□

A.4 Proof of Theorem 4.5.1

To prove the theorem, we will show that given a tournament consisting of a 3-cycle of components, the distribution over alternatives chosen by the k-RPT *oscillates* between the different components as k grows. This is made precise in the following lemma.

Lemma A.4.1. *Let A be a set of alternatives, $T \in \mathcal{T}(A)$ containing three components C_i, $i = 1, 2, 3$, such that for all alternatives $a \in C_i$ and $b \in C_{(i \bmod 3)+1}$, aTb. For $i = 1, 2, 3$ and $k \in \mathbb{N}$, denote by $p_i^{(k)}$ the probability that the k-RPT selects an alternative from C_i.*

If for some $K \in \mathbb{N}$ and $\epsilon > 0$, $p_1^{(K)} \leq \epsilon \leq 2^{-12}$, then there exists $K' > K$ such that $p_3^{(K')} \leq \epsilon/2$ and $p_2^{(K')} \geq 1 - \sqrt{\epsilon}$.

Proof. The event that some alternative from C_i is chosen by a perfect tree of height $k+1$ can be decomposed into the following two disjoint events: either an element from C_i appears at the left child of the root, and an element from C_i or $C_{(i \bmod 3)+1}$ at the right child, or an element from C_i appears at the right child and one from $C_{(i \bmod 3)+1}$ at the left. Thus, for all $k > 0$,

$$p_i^{(k+1)} = p_i^{(k)} \left(p_i^{(k)} + p_{(i \bmod 3)+1}^{(k)} \right) + p_i^{(k)} \cdot p_{(i \bmod 3)+1}^{(k)} = p_i^{(k)} \left(p_i^{(k)} + 2p_{(i \bmod 3)+1}^{(k)} \right), \tag{A.2}$$

It should be noted that (A.2) is independent of the structure of T inside the different components, but only depends on the relationship between them.

Now, consider the largest, possibly empty, set $S = \{K, K+1, K+2, \dots, \}$ such that for all $k \in S$, $p_1^{(k)} + p_2^{(k)} \leq 1/2$. It then holds for all $k \in S$ that $2p_1^{(k)} + 2p_2^{(k)} \leq 1$, and, by (A.2), that $p_1^{(k+1)} \leq p_1^{(k)} \leq p_1^{(K)} \leq 2^{-12}$; that is, $p_1^{(k)}$ is weakly decreasing for indices in S, and since we assumed $p_1^{(K)} \leq 2^{-12}$, we have that $p_1^{(k+1)} \leq 2^{-12}$ for all $k \in S$. Since $p_2^{(k)} < 0.5$ and $p_3^{(k)} \geq 0.5$, we have that for all $k \in S$, $p_2^{(k)} + 2p_3^{(k)} > 1.3$. Hence, we conclude by (A.2) that for all $k \in S$, $p_2^{(k+1)} \geq 1.3 \cdot p_2^{(k)}$.

Choosing K_1 to be the smallest integer such that $K_1 \geq K$ and $K_1 \notin S$, we have that $p_1^{(K_1)} \leq \epsilon$ and $p_3^{(K_1)} \leq 1/2$. Also, by (A.2), for all $i = 1, 2, 3$ and all $k \in \mathbb{N}$, $p_i^{(k+1)} \leq 2p_i^{(k)}$. Choosing $L \geq 12$ such that $2^{-(L+1)} \leq \epsilon \leq 2^{-L}$, we have for all $k = K_1, \dots, K_1 + \lfloor L/2 \rfloor - 1$,

$$p_1^{(k)} \leq \epsilon \cdot 2^{\lfloor L/2 \rfloor - 1} \leq \frac{2^{-\lceil L/2 \rceil}}{2} \leq \sqrt{\epsilon}/2. \tag{A.3}$$

By the assumption that $\epsilon \leq 2^{-12}$, this also implies for all such k that $p_1^{(k)} \leq 2^{-7}$.

We now claim that $K' = K_1 + \lfloor L/2 \rfloor - 1$ is as required in the statement of the lemma. Indeed, by applying (A.2), we have

$$p_3^{(K_1+1)} = p_3^{(K_1)}(p_3^{(K_1)} + 2p_1^{(K_1)}) \leq \frac{1}{2}(\frac{1}{2} + 2^{-6}) \leq 0.258,$$

and thus

$$p_3^{(K_1+2)} = p_3^{(K_1+1)}(p_3^{(K_1+1)} + 2p_1^{(K_1+1)}) \leq 0.258(0.258 + 2^{-6}) < 0.08.$$

Finally,

$$p_3^{(K_1+3)} = p_3^{(K_1+2)}(p_3^{(K_1+2)} + 2p_1^{(K_1+2)}) \leq 0.08(0.08 + 2^{-6}) < 0.0077.$$

Now, for $k = K_1 + 3, \dots, K_1 + \lfloor L/2 \rfloor - 2$, $p_3^{(k+1)} \leq p_3^{(k)}(0.0077 + 2^{-6}) < p_3^{(k)}/2^5$, since $p_3^{(k)}$ is strictly decreasing for these values of k.

It also follows directly from the above discussion that

$$p_3^{(K')} \leq p_3^{(K_1+3)} \cdot (2^{-5})^{\lfloor L/2 \rfloor - 4} \leq 2^{-5} \cdot (2^{-5})^{\lfloor L/2 \rfloor - 4} = 2^{-5\lfloor L/2 \rfloor + 15}.$$

For $L \geq 12$, $2^{-5\lfloor L/2 \rfloor + 15} \leq 2^{-(L+2)} \leq \epsilon/2$. We therefore have that $p_3^{(K')} \leq \epsilon/2$, while $p_1^{(K')} \leq \sqrt{\epsilon}/2$ by (A.3). Furthermore, since $p_2^{(K')} = 1 - (p_1^{(K')} + p_3^{(K')})$, $p_2^{(K')} \geq 1 - \sqrt{\epsilon}$. □

We will now prove a stronger version of Theorem 4.5.1.

Lemma A.4.2. *For $k \in \mathbb{N}$, denote by Δ_k the distribution corresponding to the k-RPT. Then, for every set A of alternatives, $|A| \geq 5$, there exists a tournament $T \in \mathcal{T}(A)$ such that for every $K \in \mathbb{N}$ and $\epsilon > 0$, there exists $K' \geq K$ such that*

$$\frac{\mathbb{E}_{\Gamma \sim \Delta_{K'}}[s_{\Gamma(T)}]}{\max_{i \in A} s_i} \leq \frac{1+\epsilon}{m-2}.$$

Proof of Lemma A.4.2 and Theorem 4.5.1. Let $m \geq 5$, and define a tournament as in the statement of Lemma A.4.1 with components $C_1 = \{1\}$, $C_2 = \{2\}$, and $C_3 = \{3, \ldots, m\}$, such that C_3 is transitive.

We first show that there exists K_0 such that, using the notation of Lemma A.4.1, $p_1^{(K_0)} \leq 2^{-12}$. If $m \geq 2^{12}$, this holds trivially for $K_0 = 0$, since the uniform distribution selects each alternative with probability $1/m \leq 2^{-12}$. For $m < 2^{-12}$, the claim is easily verified using a computer simulation.

Now, by Lemma A.4.1, there exists K_1 such that $p_3^{(K_1)} \leq 2^{-13}$ and $p_2^{(K_1)} \geq 1 - 2^{-6}$. Renaming the components and applying Lemma A.4.1 again, there has to exist K_2 such that $p_2^{(K_2)} \leq 2^{-14}$ and $p_1^{(K_2)} \geq 1 - 2^{-13/2}$. Another application yields K_3 satisfying $p_1^{(K_3)} \leq 2^{-15}$ and $p_3^{(K_3)} \geq 1 - 2^{-7}$. Iteratively applying the lemma in this fashion, we get that there exists $K' \geq K$ such that $p_1^{(K')} \geq 1 - \epsilon'$, for $\epsilon' = \epsilon/(m-3)$. In this case, the approximation ratio is at most

$$\frac{(1-\epsilon') + \epsilon' \cdot (m-2)}{m-2} = \frac{1+\epsilon}{m-2}. \quad \square$$

$\square$

A.5 Composition of Caterpillars

In Section 4.5 we studied the ability of randomizations over balanced trees to improve the lower bound of Section 4.4, with somewhat unexpected results. A different approach to improve the randomized lower bound is to take a tree structure that provides a good lower bound, and construct a more complex tree by composing several trees of this type to form a new structure. Since a particular randomized tree chooses alternatives according to some probability distribution, this technique is conceptually closely related to probability amplification as commonly used in the area of randomized algorithms.

In our case, the obvious candidate to be used as the basis for the composition is the RSC, both because it provides the strongest lower bound so far, and because it can conveniently be analyzed using the stationary distribution of a Markov chain. We will thus focus on *higher order caterpillar trees* obtained by replacing each leaf of a caterpillar of sufficiently large height by higher order caterpillars with order reduced by one.

To analyze the behavior of these higher order caterpillars on a particular tournament T, we again employ a Markov chain abstraction.

Given a tournament T, we inductively define Markov chains $\mathfrak{M}_k = \mathfrak{M}_k(T)$ for $k \in \mathbb{N}$ as follows: for all k, the state space of $\mathfrak{M}_k$ is A. The initial distribution and transition matrix of $\mathfrak{M}_1$ are given by those of $\mathfrak{M}$ as defined in Section 4.4.1. For $k > 1$, the initial distribution of $\mathfrak{M}_k$ is given by the stationary distribution $\pi^{(k-1)}$ of $\mathfrak{M}_{k-1}$, which can be shown to exist and be unique using similar arguments as in Section 4.4.1. Its transition matrix $P_k = P_k(T)$ is defined as

$$P_k(i,j) = \begin{cases} \pi_i^{(k-1)} + \sum_{j' : iTj'} \pi_{j'}^{(k-1)} & \text{if } i = j \\ \pi_j^{(k-1)} & \text{if } jTi \\ 0 & \text{if } iTj. \end{cases}$$

The class of tournaments used in Section 4.4.2 to show tightness of our analysis of ordinary caterpillars can also be used to show that the approximation ratio cannot be improved significantly by means of higher order caterpillars of small order. Perhaps more surprisingly, a different class of

tournaments can be shown to cause the stationary distribution of $\mathfrak{M}_k$ to oscillate as k increases, leading to a deterioration of the approximation ratio. This phenomenon is similar to the one witnessed by the proof of Theorem 4.5.1.

Theorem A.5.1. *Let A be a set of alternatives, $|A| \geq 6$, and let $K \in \mathbb{N}$. Then there exists a tournament $T \in \mathcal{T}(A)$ and $k \in \mathbb{N}$ such that $K \leq k \leq K + 5$ and the stationary distribution $\pi^{(k)}$ of $\mathfrak{M}_k(T)$ satisfies*

$$\sum_i \pi_i^{(k)} s_i \leq \frac{3}{m-2}.$$

Proof. Consider a tournament T with three components C_i, $1 \leq i \leq 3$ such that $C_i T C_j$ if $j = (i \bmod 3) + 1$ (as in the proof of Theorem 4.5.1).

For $i = 1, 2, 3$ and $k \in \mathbb{N}$, denote by $p_i^{(k)}$ the probability that an alternative from C_i is chosen from the stationary distribution of $\mathfrak{M}_k$. In particular, define $p_i^0 = |C_i|/m$. Since $p_i^{(0)} > 0$ for all i, and since T is strongly connected, $p_i^{(k)} > 0$ for all i and all $k \in \mathbb{N}$.

Then, for all $k \in \mathbb{N}$ and $i = 1, 2, 3$, and taking the subsequent index modulo three,

$$p_i^{(k+1)} = (1 - p_{i+2}^{(k)}) p_i^{(k+1)} + p_i^{(k)} p_{i+1}^{(k+1)},$$

and thus

$$p_i^{(k+1)} = \frac{p_i^{(k)}}{p_{i+2}^{(k)}} p_{i+1}^{(k+1)}.$$

Taking two steps, replacing $p_{i+1}^{(k+1)}$, and simplifying, we get

$$p_i^{(k+2)} = \frac{p_i^{(k+1)}}{p_{i+2}^{(k+1)}} p_{i+1}^{(k+2)} = \frac{p_i^{(k+1)}}{p_{i+2}^{(k+1)}} \cdot \frac{p_{i+1}^{(k+1)}}{p_i^{(k+1)}} \cdot p_{i+2}^{(k+2)} = \frac{p_i^{(k+1)} p_{i+1}^{(k)} p_{i+2}^{(k+1)} p_{i+2}^{(k+2)}}{p_{i+2}^{(k+1)} p_i^{(k)} p_i^{(k+1)}} = \frac{p_{i+1}^{(k)} p_{i+2}^{(k+2)}}{p_i^{(k)}},$$

and thus

$$\frac{p_{i+2}^{(k+2)}}{p_i^{(k+2)}} = \frac{p_i^{(k)}}{p_{i+1}^{(k)}}. \tag{A.4}$$

Analogously,

$$\frac{p_{i+1}^{(k+2)}}{p_i^{(k+2)}} = \frac{p_{i+2}^{(k)}}{p_{i+1}^{(k)}}. \tag{A.5}$$

Summing (A.4) and (A.5) and adding one,

$$\frac{p_i^{(k+2)} + p_{i+1}^{(k+2)} + p_{i+2}^{(k+2)}}{p_i^{k+2}} = \frac{p_i^{(k)} + p_{i+1}^{(k)} + p_{i+2}^{(k)}}{p_{i+1}^{(k)}}$$

and thus

$$p_i^{(k+2)} = p_{i+1}^{(k)}.$$

Choosing T such that $|C_1| = |C_2| = 1$ and $|C_3| = m - 2$, it holds for all k that

$$p_1^{(6k+4)} = p_3^{(0)} = \frac{m-2}{m}$$

and, since the sole vertex in C_1 has degree 1,

$$\sum_{i=1}^{m} \pi_i^{(6k+4)} s_i \leq \frac{m-2}{m} + \frac{2}{m} \cdot m \leq 3.$$

Observing that the sole vertex in C_2 has degree $m-2$ completes the proof. □

Appendix B

Omitted Proofs for Chapter 5

B.1 Proof of Theorem 5.4.7

Proof. It is obvious that TREE-SAT$\in \mathcal{NP}$. In order to show $\mathcal{NP}$-hardness, we present a reduction from 3SAT. In this problem, one is given a conjunction of clauses, where each clause is a disjunction of three literals. One is asked whether the given formula has a satisfying assignment. It is common knowledge that 3SAT is $\mathcal{NP}$-complete.

Given an instance of 3SAT with variables $\{x_1, \ldots, x_m\}$, and clauses $\{l_1^j \vee l_2^j \vee l_3^j\}_{j=1}^k$, we construct an instance of TREE-SAT as follows: the set of alternatives is

$$A = \{a, b, x_1, \neg x_1, c_1, x_2, \neg x_2, c_2, \ldots, x_m, \neg x_m, c_m\}.$$

For each clause j, we define a tournament T_j as some tournament that satisfies the following restrictions:

1. l_1^j, l_2^j and l_3^j beat any other alternative among the alternatives $x_i, \neg x_i$.
2. a loses to l_1^j, l_2^j and l_3^j, but beats any other alternative among the alternatives $x_i, \neg x_i$.

In addition, all tournaments in our instance of TREE-SAT satisfy the following conditions:

1. b beats any alternative which corresponds to a literal, but loses to a.
2. For all $i = 1, \ldots, m$, $\neg x_i$ beats x_i.
3. c_i loses to x_i and $\neg x_i$, and beats any other alternative.

Finally, for each tournament, we require the winner to be alternative b. We now proceed to construct the given (partially assigned) tree. We start, as in the proof of Theorem 5.4.1, by defining a gadget which we call a *literal gadget*, illustrated in Figure B.1.

In this subtree, two leaves are already assigned with x_i and c_i. Now, with respect to any of the tournaments we defined, if we assign $\neg x_i$ to the last leaf, then $\neg x_i$ proceeds to beat c_i, and subsequently beats x_i. If we assign x_i to the third leaf, then x_i beats c_i and wins the election. If we assign any other alternative, that alternative is defeated by c_i, which in turn is beaten by x_i. To conclude the point, either x_i or $\neg x_i$ survives the gadget; $\neg x_i$ survives iff it is assigned to the third leaf.

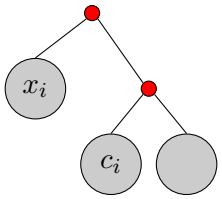


Figure B.1: Literal gadget.

Given these literal gadgets, we can assume without loss of generality that we can construct a tree such that in some of the leaves the only possible assignments are x_i or $\neg x_i$; we shall mark these leaves by $x_i : \neg x_i$. The complete (partially labeled) tree in the constructed Tree-SAT instance is described in Figure B.2.

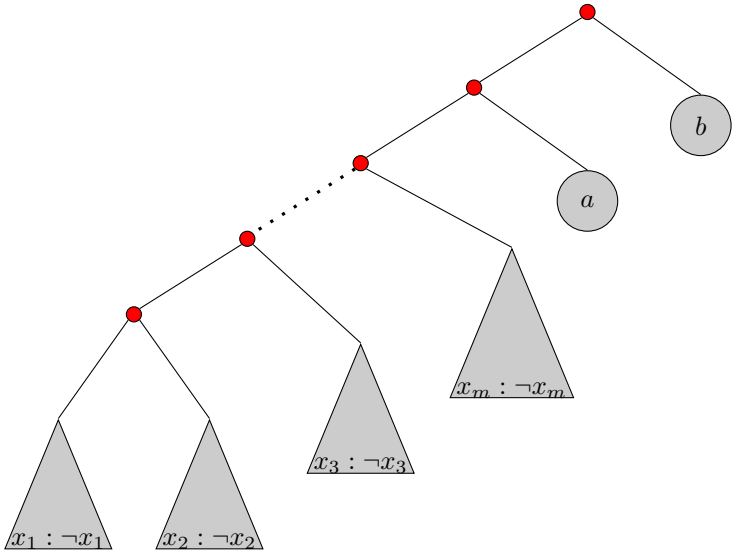


Figure B.2: The reduction.

We now prove that this is indeed a reduction. We first have to show that if the given 3SAT instance is satisfiable, there is an assignment to the leaves of our tree (i.e., choices of x_i or $\neg x_i$) such that, for each of the m tournaments, the winner is b. Consider some satisfying assignment to the 3SAT instance, and apply the assignment to the above tree. Now, consider some tournament T_j. At least one of the literals l_1^j, l_2^j or l_3^j must be true; as these three literals beat all other literals in the tournament T_j, one of these three literals reaches the competition versus a, and wins; subsequently, this literal loses to alternative b. Therefore, b is the winner of the election. Since this is true for any $j = 1, \ldots, m$, we have that the assignment is consistent with the given tournaments.

In the other direction, consider an instance of 3SAT which is not satisfiable. Fix some assignment to the leaves of the tree; the corresponding assignment to the 3SAT instance is not satisfying. Therefore, there is some j such that l_1^j, l_2^j, and l_3^j are all false. This implies that in T_j some other literal other than these three reaches the top of the tree to compete against a, and loses. Subsequently, a competes against b and wins, making a the winner of the election with respect to

tournament T_j. Hence, this is not an assignment which is consistent with all tournaments—but this is true with respect to any assignment. □

Appendix C

Omitted Proofs and Results for Chapter 6

C.1 Proof of Theorem 6.4.2

Proof. We shall first prove the theorem for the case when $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$ (Steps 1 and 2), and then extend the result to homogeneous linear functions over $\mathbb{R}$ (Step 3). We have already shown truthfulness, and therefore directly turn to approximate efficiency. In the following, we denote the empirical risk minimizer by $f^*(x) \equiv a^*$, and the function returned by project-and-fit by $f(x) \equiv a$.

Step 1: $|\{\, y_{ij} :\ y_{ij} \leq a \,\}| \geq \frac{1}{4}nm$ and $|\{\, y_{ij} :\ y_{ij} \geq a \,\}| \geq \frac{1}{4}nm$. Let $\tilde{y}_{ij}$ denote the projected values of agent i. As noted above, when $\mathcal{F}$ is the class of constant functions, the mechanism in fact returns the median of the values $\tilde{y}_{ij}$, and thus

$$|\{\, \tilde{y}_{ij} :\ \tilde{y}_{ij} \leq a \,\}| \geq \frac{1}{2}nm \quad . \tag{C.1}$$

Furthermore, since for all j, $\tilde{y}_{ij}$ is the median of the original values y_{ij} of agent i, it must hold that at least half of these values are smaller than their corresponding original value, i.e.

$$|\{\, y_{ij} :\ y_{ij} \leq a \,\}| \geq \frac{1}{2}|\{\, \tilde{y}_{ij} :\ \tilde{y}_{ij} \leq a \,\}| \quad . \tag{C.2}$$

Combining Equations C.1 and C.2, we obtain $|\{\, y_{ij} :\ y_{ij} \leq a \,\}| \geq \frac{1}{4}nm$. By symmetrical arguments, we get that $|\{\, y_{ij} :\ y_{ij} \geq a \,\}| \geq \frac{1}{4}nm$.

Step 2: 3-efficiency for constant functions. Denote $d = |a - a^*|$, and assume without loss of

generality that $a < a^*$. We now have that

$$\begin{aligned}
\hat{\text{risk}}(f,S) &= \frac{1}{nm}\sum_{i,j}|y_{ij}-a| \\
&= \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(a-y_{ij}) + \sum_{i,j:a<y_{ij}\le a^*}(y_{ij}-a) + \sum_{i,j:y_{ij}>a^*}(y_{ij}-a)\right) \\
&\le \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(a-y_{ij}) + \sum_{i,j:a<y_{ij}\le a^*} d + \sum_{i,j:y_{ij}>a^*}(d+(y_{ij}-a^*))\right) \\
&= \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(a-y_{ij}) + \sum_{i,j:y_{ij}>a^*}(y_{ij}-a^*) + |\{\, i,j : y_{ij} > a \,\}|\cdot d\right) \quad .
\end{aligned}$$

We now bound the last expression above by replacing $|\{i,j:\ y_{ij} > a\}|$ with its upper bound $\frac{3}{4}nm$ derived in Step 1 and obtain

$$\hat{\text{risk}}(f,S) \le \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(a-y_{ij}) + \sum_{i,j:y_{ij}>a^*}(y_{ij}-a^*) + \frac{3}{4}nm\cdot d\right) \quad .$$

Similarly,

$$\hat{\text{risk}}(f^*,S) \ge \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(d+(a-y_{ij})) + \sum_{i,j:y_{ij}>a^*}(y_{ij}-a^*)\right) \quad ,$$

and using Step 1,

$$\hat{\text{risk}}(f^*,S) \ge \frac{1}{nm}\left(\sum_{i,j:y_{ij}\le a}(a-y_{ij}) + \sum_{i,j:y_{ij}>a^*}(y_{ij}-a^*) + \frac{1}{4}nm\cdot d\right) \quad .$$

Since two of the expressions in the upper bound for $\hat{\text{risk}}(f,S)$ and the lower bound for $\hat{\text{risk}}(f^*,S)$ are identical, it is now self-evident that $\hat{\text{risk}}(f,S)/\hat{\text{risk}}(f^*,S) \le 3$.

Step 3: Extension to homogeneous linear functions over $\mathbb{R}$. We describe a reduction from the case of homogeneous functions over $\mathbb{R}$ to the case of constant functions over $\mathbb{R}$. Given a sample S, we create a sample S' by mapping each example $(x,y) \in S$ to $|x|$ copies of the example $(x, y/x)$.[1] Let f_1 be the homogeneous linear function defined by $f_1(x) = a \cdot x$, and let f_2 be the constant function defined by $f_2(x) = a$. It is now straightforward to show that $\hat{\text{risk}}(f_1,S) = \hat{\text{risk}}(f_2,S')$, and that project-and-fit chooses f_1 when given the class of homogeneous linear functions and S if and only if it chooses f_2 when given the class of constant functions and S'. □

C.2 Proof of Theorem 6.4.3

We first require a technical result. For this, assume that $\mathcal{F}$ is the class of constant functions over $\mathbb{R}^k$, let $N = \{1,2\}$, and fix some truthful mechanism M.

[1]Here we assume that the values x are integers, but it is possible to deal with noninteger values by assigning weights.

Lemma C.2.1. *Let $q, t \in \mathbb{N}$, and define $m = 2t - 1$. Then there exists a sample S defined by*

$$S_1 = \{(\mathbf{x}_{11}, y), (\mathbf{x}_{12}, y), \ldots, (\mathbf{x}_{1m}, y)\} \quad \textit{and}$$
$$S_2 = \{(\mathbf{x}_{21}, y'), (\mathbf{x}_{22}, y'), \ldots, (\mathbf{x}_{2m}, y')\},$$

such that $y - y' = 2^q$ and $M(S) \geq y - \frac{1}{2}$ or $M(S) \leq y' + \frac{1}{2}$.

Proof. We perform an induction on q. For $q = 0$, we simply set $y = 1$ and $y' = 0$. Now, let S be a sample as in the formulation of the lemma, and let $a = M(S)$, i.e. a is the constant function returned by M given S. We distinguish two different cases.

Case 1: If $a \geq y - 1/2$, let S' such that $S'_1 = S_1$ and

$$S'_2 = \{(\mathbf{x}_{21}, 2y' - y), \ldots, (\mathbf{x}_{2m}, 2y' - y)\} \quad .$$

Notice that $y - (2y' - y) = 2(y - y')$, so the distance between the values has doubled. Denote $a' = M(S')$. Due to truthfulness of M, it must hold that $\ell(a', y') \geq \ell(a, y') \geq 2^q - \frac{1}{2}$. Otherwise, if agent 2's true type was S_2, he would benefit by saying that his type is in fact S'_2. Therefore, $a' \geq y - \frac{1}{2}$ or $a' \leq y' - (2^q + \frac{1}{2}) = 2y' - y + \frac{1}{2}$.

Case 2: If $a \leq y' + \frac{1}{2}$, let S' such that $S'_2 = S_2$ and

$$S'_1 = \{(\mathbf{x}_{11}, 2y - y'), \ldots, (\mathbf{x}_{1m}, 2y - y')\} \quad .$$

Analogously to Case 1, the induction step follows from truthfulness of M with respect to agent 1. □

Proof of Theorem 6.4.3. Consider the sample S as in the statement of the lemma, and assume without loss of generality that $M(S) = a \geq y - \frac{1}{2}$. Otherwise, symmetrical arguments apply. We first observe that if M is approximately efficient, it cannot be the case that $M(S) > y$. Otherwise, let S' be the sample such that $S'_1 = S_1$ and

$$S'_2 = \{(\mathbf{x}_{21}, y), \ldots, (\mathbf{x}_{2m}, y)\} \quad ,$$

and denote $a' = M(S')$. Then, by truthfulness with respect to agent 2, $\ell(a', y') \geq \ell(a, y')$. It follows that $a' \neq y$, and therefore $\hat{\text{risk}}(a', S') > 0$. Since $\hat{\text{risk}}(y, S') = 0$, the efficiency ratio is not bounded.

Now let S'' be such that $S''_2 = S_2$, and

$$S''_1 = \{(\mathbf{x}_{11}, y), (\mathbf{x}_{12}, y), \ldots, (\mathbf{x}_{1t}, y), (\mathbf{x}_{1,t+1}, y'), \ldots, (\mathbf{x}_{1m}, y')\} \quad ,$$

i.e. agent 1 has t points at y and $t - 1$ points at y'. Let $a'' = M(S'')$. Due to truthfulness, it must hold that $\ell(a'', y) = \ell(a, y)$, since agent 1's empirical risk minimizer with respect to both S and S'' is y. Since we already know that $y - \frac{1}{2} \leq a \leq y$, we get that $a'' \geq y - \frac{1}{2}$, and thus $\hat{\text{risk}}(a'', S'') \geq \frac{(3t-2)}{(4t-2)}(2^q - \frac{1}{2})$. On the other hand, the empirical risk minimizer on S'' is y', and $\hat{\text{risk}}(y', S'') \leq \frac{t}{4t-2} 2^q$. The efficiency ratio $\hat{\text{risk}}(a'', S'')/\hat{\text{risk}}(y', S'')$ tends to 3 as t and q tend to infinity.

We will now explain how this result can be extended to homogeneous linear functions over $\mathbb{R}^k$. For this, define the sample S by

$$S_1 = \{\langle t - 1, 0, \ldots, 0\rangle, (t-1)y), (\langle t, 0, \ldots, 0\rangle, ty)\} \qquad \text{and}$$
$$S_2 = \{\langle t - 1, 0, \ldots, 0\rangle, (t-1)y'), (\langle t, 0, \ldots, 0\rangle, ty')\} \quad .$$

As with constant functions, a homogeneous linear function defined by $\mathbf{a}$ satisfies $\hat{\text{risk}}(\mathbf{a}, S_1) = |a_1 - y|$, and $\hat{\text{risk}}(\mathbf{a}, S_2) = |a_1 - y'|$. Therefore, we can use similar arguments to the ones above to show that there exists a sample S with $y - y' = 2^q$, and if $\mathbf{a} = M(S)$ for some truthful mechanism M, then $y - \frac{1}{2} \leq a_1 \leq y$ or $y' \leq a_1 \leq y' + \frac{1}{2}$. As before, we complete the proof by splitting the points controlled by agent 1, i.e. by considering the sample S' where $S'_1 = \{\langle t-1, 0, \ldots, 0\rangle, (t-1)y'), (\langle t, 0, \ldots, 0\rangle, ty)\}$. □

C.3 Justification of Conjecture 6.4.5

In order to justify the conjecture, it will be instructive to once again view the hypothesis class $\mathcal{F}$ as a set of alternatives. The agents' types induce a preference order over this set of alternatives. Explicitly, agent i weakly prefers function f_1 to function f_2 if and only if $\hat{\text{risk}}(f_1, S_i) \leq \hat{\text{risk}}(f_2, S_i)$. A mechanism without payments is a social choice function from the agents' preferences over $\mathcal{F}$ to $\mathcal{F}$.

Recall that the Gibbard-Satterthwaite Theorem (Theorem 2.4.2) asserts that every truthful social choice function from the set of *all* linear preferences over some set A of alternatives to A must be *dictatorial*, in the sense that there is some agent d such that the social outcome is always the one most preferred by d. Observe that this theorem does not directly apply in our case, since agents' preferences are restricted to a strict subset of all possible preference relations over $\mathcal{F}$.

For the time being, let us focus on homogeneous linear functions f over $\mathbb{R}^k$, $k \geq 2$. This class is isomorphic to $\mathbb{R}^k$, as every such function can be represented by a vector $\mathbf{a} \in \mathbb{R}^k$ such that $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x}$. Let R be a weak preference relation over $\mathbb{R}^k$, and let P be the asymmetric part of R (i.e. $\mathbf{a} P \mathbf{a}'$ if and only if $\mathbf{a} R \mathbf{a}'$ and not $\mathbf{a}' R \mathbf{a}$). R is called *star-shaped* if there is a unique point $\mathbf{a}^* \in \mathbb{R}^k$ such that for all $\mathbf{a} \in \mathbb{R}^k$ and $\lambda \in (0,1)$, $\mathbf{a}^* P (\lambda \mathbf{a}^* + (1-\lambda)\mathbf{a}) P \mathbf{a}$. In our case preferences are clearly star-shaped, as for any $\mathbf{a}, \mathbf{a}' \in \mathbb{R}^k$ and any sample S, $\hat{\text{risk}}((\lambda \mathbf{a} + (1-\lambda)\mathbf{a}'), S) = \lambda \hat{\text{risk}}(\mathbf{a}, S) + (1-\lambda)\hat{\text{risk}}(\mathbf{a}', S)$.

A preference relation R over $\mathbb{R}^m$ is called *separable* if for every j, $1 \leq j \leq m$, all $x, y \in \mathbb{R}^m$, and all $a_j, b_j \in \mathbb{R}$,

$$\langle x_{-j}, a_j \rangle \, R \, \langle x_{-j}, b_j \rangle \quad \text{if and only if} \quad \langle y_{-j}, a_j \rangle \, R \, \langle y_{-j}, b_j \rangle \quad ,$$

where $\langle x_{-j}, a_j \rangle = \langle x_1, \ldots, x_{j-1}, a_j, x_{j+1}, \ldots, x_m \rangle$. The following example establishes that in our setting preferences are not separable.

Example C.3.1. Let $\mathcal{F}$ be the class of homogeneous linear functions over $\mathbb{R}$, and define $S_1 = \{(\langle 1, 1\rangle, 0)\}$. Then agent 1 prefers $\langle -1, 1\rangle$ to $\langle -1, 2\rangle$, but also prefers $\langle -2, 2\rangle$ to $\langle -2, 1\rangle$.

Border and Jordan [15] investigate a setting where the set of alternatives is $\mathbb{R}^k$. They give possibility results for the case when preferences are star-shaped and separable. On the other hand, when $k \geq 2$ and the separability criterion is slightly relaxed, in a way which we will not elaborate on here, then any truthful social choice function must necessarily be dictatorial.

Border and Jordan's results also require surjectivity: the social choice function has to be onto $\mathbb{R}^k$.[2] While this is a severe restriction in general, it is in fact very natural in our context. If all agents have values consistent with some function f, then the mechanism can have a bounded efficiency ratio only if its output is the function f (indeed, f has loss 0, while any other function has strictly positive loss). Therefore, any approximately efficient mechanism must be surjective.

[2] Border and Jordan [15] originally required unanimity, but their theorems can be reformulated using surjectivity [142].

The above discussion leads us to believe that there is no truthful approximation mechanism for homogeneous linear functions over $\mathbb{R}^k$ for any $k \geq 2$. Conjecture 6.4.5 simply formalized this statement.

Bibliography

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the 37th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 684–693, 2005.

[2] N. Alon. Ranking tournaments. *SIAM Journal of Discrete Mathematics*, 20(1–2):137–142, 2006.

[3] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley and Sons, 1992.

[4] Y. Bachrach, E. Markakis, A. D. Procaccia, J. S. Rosenschein, and A. Saberi. Approximating power indices. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 943–950, 2008.

[5] E. Baharad and Z. Neeman. The asymptotic strategyproofness of scoring and Condorcet consistent rules. *Review of Economic Design*, 4:331–340, 2002.

[6] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 1st ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 16–25, 2006.

[7] J. Bartholdi and J. Orlin. Single Transferable Vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

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

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

[10] J. Bartholdi, C. A. Tovey, and M. A. Trick. How hard is it to control an election. *Mathematical and Computer Modelling*, 16:27–40, 1992.

[11] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

[12] E. Beigman and R. Vohra. Learning from revealed preference. In *Proceedings of the 7th ACM Conference on Electronic Commerce (ACM-EC)*, pages 36–42, 2006.

[13] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. In *Proceedings of the 11th Scandinavian Workshop on Algorithm Theory (SWAT)*, 2008.

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

[15] K. Border and J. Jordan. Straightforward elections, unanimity and phantom voters. *Review of Economic Studies*, 50:153–170, 1983.

[16] S. Brams, D. M. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15:211–236, 1998.

[17] F. Brandt and F. Fischer. Computing the Minimal Covering Set. *Mathematical Social Sciences*, 2008. To appear.

[18] F. Brandt, F. Fischer, and P. Harrenstein. The computational complexity of choice sets. In *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 82–91, 2007.

[19] F. Brandt, F. Fischer, P. Harrenstein, and M. Mair. A computational analysis of the Tournament Equilibrium Set. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 38–43, 2008.

[20] N. H. Bshouty, N. Eiron, and E. Kushilevitz. PAC learning with nasty noise. *Theoretical Computer Science*, 288(2):255–275, 2002.

[21] I. Caragiannis, J. A. Covey, M. Feldman, C. M. Homan, C. Kaklamanis, N. Karanikolas, A. D. Procaccia, and J. S. Rosenschein. On the approximability of Dodgson and Young elections. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1058–1067, 2009.

[22] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[23] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[24] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to Computational Social Choice. In *SOFSEM 2007: Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer-Verlag, 2007.

[25] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[26] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 613–619, 2006.

[27] V. Conitzer. Eliciting single-peaked preferences using comparison queries. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 408–415, 2007.

[28] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategyproofness. In *Proceedings of the 18th AAAI Conference on Artificial Intelligence (AAAI)*, pages 392–397, 2002.

[29] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 103–110, 2002.

[30] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, 2003.

[31] V. Conitzer and T. Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 128–135, 2004.

[32] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM-EC)*, pages 78–87, 2005.

[33] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 627–634, 2006.

[34] V. Conitzer, A. Davenport, and H. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 620–626, 2006.

[35] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.

[36] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 776–782, 2006.

[37] P. J. Coughlan and M. Le Breton. A social choice function implementable via backward induction with values in the ultimate uncovered set. *Review of Economic Design*, 4:153–160, 1999.

[38] A. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence (AAAI)*, pages 697–702, 2004.

[39] O. Dekel, F. Fischer, and A. D. Procaccia. Incentive compatible regression learning. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 277–286, 2008.

[40] S. Dobzinski and A. D. Procaccia. Frequent manipulability of elections: The case of two voters. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, pages 653–664, 2008.

[41] B. Dutta and A. Sen. Implementing generalized Condorcet social choice functions via backward induction. *Social Choice and Welfare*, 10:149–160, 1993.

[42] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1987.

[43] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science (LNCS)*, pages 206–215. Springer-Verlag, 2005.

[44] E. Elkind and H. Lipmaa. Small coalitions cannot manipulate voting. In *Financial Cryptography and Data Security*, volume 3570 of *Lecture Notes in Computer Science (LNCS)*, pages 285–297. Springer-Verlag, 2005.

[45] E. Ephrati and J. S. Rosenschein. A heuristic technique for multiagent planning. *Annals of Mathematics and Artificial Intelligence*, 20:13–67, 1997.

[46] G. Erdélyi, L. A. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. In *Fundamentals of Computation Theory*, volume 4639 of *Lecture Notes in Computer Science (LNCS)*, pages 300–311. Springer-Verlag, 2007.

[47] P. Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1569–1572, 2008.

[48] P. Faliszewski, E. Hemaspaandra, , and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 641–646, 2006.

[49] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 724–730, 2007.

[50] P. Faliszewski, E. Hemaspaandra, , and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 983–990, 2008.

[51] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM)*, pages 165–176, 2008.

[52] R. Farquharson. *Theory of Voting*. Yale University Press, 1969.

[53] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):643–652, 1998.

[54] W. Feller. *Introduction to Probability Theory and its Applications*, volume 1, page 254. John Wiley, 3rd edition, 1968.

[55] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *The Annals of Applied Probablity*, 1(1):62–87, 1991.

[56] F. Fischer, A. D. Procaccia, and A. Samorodnitsky. A new perspective on implementation by voting trees. In *Proceedings of the 10th ACM Conference on Electronic Commerce (ACM-EC)*, 2009. To appear.

[57] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 243–249, 2008.

[58] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[59] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of a recommender system. In *Proceedings of the 3rd Annual Conference on Autonomous Agents (AGENTS)*, pages 434–435, 1999.

[60] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[61] S. A. Goldman and R. H. Sloan. Can PAC learning algorithms tolerate random attribute noise? *Algorithmica*, 14(1):70–84, 1995.

[62] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[63] D. Haussler. Decision theoretic generalization of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[64] T. Haynes, S. Sen, N. Arora, and R. Nadella. An automated meeting scheduling system that utilizes user preferences. In *Proceedings of the 1st Annual Conference on Autonomous Agents (AGENTS)*, pages 308–315, 1997.

[65] E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[66] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[67] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[68] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1308–1314, 2007.

[69] M. Herrero and S. Srivastava. Decentralization by multistage voting procedures. *Journal of Economic Theory*, 56:182–201, 1992.

[70] C. Homan and L. A. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson election winners. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 528–539, 2006.

[71] O. Hudry. A note on "Banks winners in tournaments are difficult to recognize" by G. J. Woeginger. *Social Choice and Welfare*, 23:113–114, 2004.

[72] O. Hudry. Improvements of a branch and bound method to compute the Slater orders of tournaments. Technical report, ENST, 2006.

[73] K. Jogdeo and S. Samuels. Monotone convergence of binomial probabilities and a generalization of Ramanujan's equation. *Annals of Mathematical Statistics*, 39:1191–1195, 1968.

[74] J. Kahn, M. Saks, and D. Sturtevant. A topological approach to evasiveness. *Combinatorica*, 4:297–306, 1984.

[75] G. Kalai. Learnability and rationality of choice. *Journal of Economic Theory*, 113(1):104–117, 2003.

[76] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

[77] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 95–103, 2007.

[78] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[79] V. King. Lower bounds on the complexity of graph properties. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 468–476, 1988.

[80] C. Klamler. The Dodgson ranking and its relation to Kemeny's method and Slater's rule. *Social Choice and Welfare*, 23(1):91–102, 2004.

[81] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the 2nd Multidisciplinary Workshop on Advances in Preference Handling (M-PREF)*, 2005.

[82] G. Laffond, J. F. Laslier, and M. Le Breton. The Copeland measure of Condorcet choice functions. *Discrete Applied Mathematics*, 55:273–279, 1994.

[83] S. Lahaie and D. C. Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce (ACM-EC)*, pages 180–188, 2004.

[84] J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.

[85] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1366–1371, 2007.

[86] J. Lang, M.-S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Winner determination in sequential majority voting. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1372–1377, 2007.

[87] J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer, 1997.

[88] D. Lehmann, L. I. O'Callaghan, and Y. Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002.

[89] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[90] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

[91] N. Littlestone. Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings of the 4th Annual Workshop on Computational Learning Theory (COLT)*, pages 147–156, 1991.

[92] J. C. McCabe-Dansted. Approximability and computational feasibility of Dodgson's rule. Master's thesis, University of Auckland, 2006.

[93] J. C. McCabe-Dansted, G. Pritchard, and A. M. Slinko. Approximability of Dodgson's rule. In *Proceedings of the 1st International Workshop on Computational Social Choice (COMSOC)*, pages 331–344, 2006.

[94] D. C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21: 608–610, 1953.

[95] R. D. McKelvey and R. G. Niemi. A multistage game representation of sophisticated voting for binary procedures. *Journal of Economic Theory*, 18:1–22, 1978.

[96] R. Meir, A. D. Procaccia, and J. S. Rosenschein. A broader picture of the complexity of strategic behavior in multi-winner elections. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 991–998, 2008.

[97] R. Meir, A. D. Procaccia, and J. S. Rosenschein. Strategyproof classification under constant hypotheses: A tale of two functions. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 126–131, 2008.

[98] N. Miller. A new solution set for tournaments and majority voting: Further graph theoretical approaches to the theory of voting. *Americal Journal of Political Science*, 24:68–96, 1980.

[99] B. L. Monroe. Fully proportional representation. *American Political Science Review*, 89(4): 925–940, 1995.

[100] J. W. Moon. *Topics on Tournaments*. Holt, Reinhart and Winston, 1968.

[101] H. Moulin. Generalized Condorcet-winners for single peaked and single-plateau preferences. *Social Choice and Welfare*, 1(2):127–147, 1984.

[102] H. Moulin. Choosing from a tournament. *Social Choice and Welfare*, 3:271–291, 1986.

[103] B. K. Natarajan. *Machine Learning: A Theoretical Approach*. Morgan Kaufmann, 1991.

[104] N. Nisan. Introduction to mechanism design (for computer scientists). In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 9. Cambridge University Press, 2007.

[105] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35 (1–2):166–196, 2001.

[106] K. Oflazer and G. Tür. Morphological disambiguation by voting constraints. In *Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 222–229, 1997.

[107] B. Peleg and A. D. Procaccia. Mediators enable truthful voting. Discussion paper 451, Center for the Study of Rationality, The Hebrew University of Jerusalem, 2007.

[108] B. Peleg and A. D. Procaccia. Implementation by mediated equilibrium. Discussion paper 466, Center for the Study of Rationality, The Hebrew University of Jerusalem, 2007.

[109] D. Pennock, E. Horvitz, and L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th AAAI Conference on Artificial Intelligence (AAAI)*, pages 729–734, 2000.

[110] J. Perote and J. Perote-Peña. Strategy-proof estimators for simple regression. *Mathematical Social Sciences*, 47:153–176, 2004.

[111] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1464–1469, 2007.

[112] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

[113] A. D. Procaccia. Towards a theory of incentives in machine learning. *SIGecom Exchanges*, 7 (2), 2008.

[114] A. D. Procaccia. A note on the query complexity of the Condorcet winner problem. *Information Processing Letters*, 2008. To appear.

[115] A. D. Procaccia and J. S. Rosenschein. Extensive-form argumentation games. In *Proceedings of the 3rd European Workshop on Multi-Agent Systems (EUMAS)*, pages 312–322, 2005.

[116] A. D. Procaccia and J. S. Rosenschein. The distortion of cardinal preferences in voting. In *Proceedings of the 10th International Workshop on Cooperative Information Agents (CIA)*, volume 4149 of *Lecture Notes in Computer Science (LNCS)*, pages 317–331. Springer-Verlag, 2006.

[117] A. D. Procaccia and J. S. Rosenschein. The communication complexity of coalition formation among autonomous agents. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 505–512, 2006.

[118] A. D. Procaccia and J. S. Rosenschein. Learning to identify winning coalitions in the PAC model. pages 673–675, 2006.

[119] A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.

[120] A. D. Procaccia and J. S. Rosenschein. Average-case tractability of manipulation in elections via the fraction of manipulators. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 718–720, 2007.

[121] A. D. Procaccia and J. S. Rosenschein. A computational characterization of multiagent games with fallacious rewards. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1152–1159, 2007.

[122] A. D. Procaccia, Y. Bachrach, and J. S. Rosenschein. Gossip-based aggregation of trust in decentralized reputation systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1470–1475, 2007.

[123] A. D. Procaccia, M. Feldman, and J. S. Rosenschein. Approximability and inapproximability of dodgson and young elections. Discussion paper 466, Center for the Study of Rationality, The Hebrew University of Jerusalem, 2007.

[124] A. D. Procaccia, J. S. Rosenschein, and G. A. Kaminka. On the robustness of preference aggregation in noisy environments. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 416–422, 2007.

[125] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1476–1481, 2007.

[126] A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. Learning voting trees. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 110–115, 2007.

[127] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. On the complexity of achieveing proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[128] A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. The learnability of voting rules. *Artificial Intelligence*, 2009. In press.

[129] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 475–484, 1997.

[130] R. Rivest and S. Vuillemin. On recognizing graph properties from adjacency matrices. *Theoretical Computer Science*, 3:371–384, 1976.

[131] A. L. Rosenberg. The time required to recognize properties of graphs: A problem. *SIGACT News*, 5(4):15–16, 1973.

[132] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.

[133] M. Rothkopf. Thirteen reasons the Vickrey-Clarke-Groves process is not practical. *Operations Research*, 55(2):191–197, 2007.

[134] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1–2):209–238, 1999.

[135] M. 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.

[136] J. Schummer and R. V. Vohra. Mechanism design without money. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 10. Cambridge University Press, 2007.

[137] I. Segal. The communication requirements of social choice rules and supporting budget sets. *Journal of Economic Theory*, 136:341–378, 2007.

[138] J. Shawe-Taylor and N. Cristianini. *Support Vector Machines and other Kernel Based Learning Methods*. Cambridge University Press, 2000.

[139] G. Sigletos, G. Paliouras, C. Spyropoulos, and M. Hatzopoulos. Combining information extractions systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6:1751–1782, 2005.

[140] A. Sinclair and M. Jerrum. Approximate counting, uniform generation, and rapidly mixing Markov chains. *Information and Computation*, 82:93–133, 1989.

[141] A. Slinko. How large should a coalition be to manipulate an election? *Mathematical Social Sciences*, 47(3):289–293, 2004.

[142] Y. Sprumont. Strategyproof collective choice in economic and political environments. *The Canadian Journal of Economics*, 28(1):68–107, 1995.

[143] S. Srivastava and M. A. Trick. Sophisticated voting rules: The case of two tournaments. *Social Choice and Welfare*, 13:275–289, 1996.

[144] L. Trevisan. Lecture notes on computational complexity, 2002. Lecture 12.

[145] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[146] W. Vickrey. Counter speculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[147] L. Xia and V. Conitzer. Generalized Scoring Rules and the frequency of coalitional manipulability. In *Proceedings of the 9th ACM Conference on Electronic Commerce (ACM-EC)*, pages 109–118, 2008.

[148] L. Xia and V. Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the 9th ACM Conference on Electronic Commerce (ACM-EC)*, pages 99–108, 2008.

[149] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 196–201, 2008.

[150] L. Xia, J. Lang, and M. Ying. Sequential voting rules and multiple elections paradoxes. In *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 279–288, 2007.

[151] L. Xia, J. Lang, and M. Ying. Strongly decomposable voting rules on multiattribute domains. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 776–781, 2007.

[152] L. Xia, V. Conitzer, A. D. Procaccia, and J. S. Rosenschein. Complexity of unweighted manipulation under some common voting rules. In *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC)*, 2008. To appear.

[153] A. C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of the 17th Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, 1977.

[154] H. P. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16:335–353, 1977.

[155] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 277–286, 2008.