Machine Learning and Multiagent Preferences

Ritesh Noothigattu

August 2020
CMU-ML-20-109

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee
Ariel D. Procaccia (Chair), Harvard University
Maria-Florina Balcan, Carnegie Mellon University
Nihar B. Shah, Carnegie Mellon University
Milind Tambe, Harvard University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation award numbers CCF1525932 and IIS1714140, the Office of the Naval Research award number N000141712428, and the United States Army Research Office award number W911NF1320045.

Keywords: machine learning, multiagent preferences, fairness, social choice, axioms, pooling, reinforcement learning.

To everyone who helps me keep my smile :)

Abstract

One of the most well known settings dealing with multiagent preferences is voting and social choice. In classical social choice, each of the n agents presents a ranking over the m candidates, and the goal is to find a winning candidate (or a consensus ranking) that is the most "fair" outcome. In this thesis, we consider several variants of this standard setting. For instance, the domain may have an uncountably infinite number of alternatives, and we need to learn each voter's preferences from a few pairwise comparisons among them. Or, we have a markov decision process, and each voter's preferences are represented by its reward function. Can we find a consensus policy that everyone would be happy with? Another example is the setting of a conference peer review system, where the agents are the reviewers, and their preferences are given by the defining characteristics they use to accept a paper. Our goal is then to use these preferences to make consensus decisions for the entire conference. We also consider the setting where agents have utility functions over a given set of outcomes, and our goal is to learn a classifier that is fair with respect to these preferences. Broadly speaking, this thesis tackles problems in three areas: (i) fairness in machine learning, (ii) voting and social choice, and (iii) reinforcement learning, each of them handling multiagent preferences with machine learning.

Acknowledgments

Without a doubt, I have been extremely fortunate to be advised by Ariel Procaccia. Even though it sounds obligatory, this has been absolutely true in my case. Ariel has been extremely supportive and very encouraging throughout my PhD studies. I literally could not ask for a better adviser. My first encounter with Ariel was during my MLD orientation, where Ariel presented his research, and I was really captivated by his talk, which was a perfect blend of discrete math and probability/statistics, and I ultimately decided to choose him as my adviser. Right from the beginning, Ariel has always used positive encouragement to motivate me, and I have never seen him be negative or forceful. His extreme niceness, friendliness and positivity definitely motivated me to push further, and I was also able to openly share my thoughts with him. His support has helped me achieve much more than I would have otherwise. Ariel is also an amazing perfectionist, which helped me improve in almost every aspect of research. For instance, I clearly remember when I was writing some of the sections of my first papers with him, he spent substantial time in proofreading and updating them, and then gave me a list of points on how I could improve my writing. Looking at the extremely well-written and polished "after" versions of these sections, and his feedback, has been the main reason my writing improved several fold since then. I feel extremely fortunate to be advised by Ariel, and if I were given the option to go back in time and pick an adviser again, I would definitely choose him again without a doubt. And, if I ever become an adviser myself, I would aspire to be at least half as good as him :)

I would also like to thank Maria-Florina Balcan, Nihar B. Shah, and Milind Tambe for being part of my thesis committee, and also all my other collaborators who helped make this thesis possible: Edmond Awad, Djallel Bouneffouf, Murray Campbell, Allissa Chan, Rachita Chandra, Travis Dick, Sohan Dsouza, Snehalkumar S. Gaikwad, Nika Haghtalab, Anson Kahng, Ji Tae Kim, Daniel Kusbit, Min Kyung Lee, Siheon Lee, Piyush Madan, Leandro S. Marcolino, Nicholas Mattei, Dominik Peters, Christos-Alexandros Psomas, Iyad Rahwan, Pradeep Ravikumar, Eric Rice, Francesca Rossi, Daniel See, Moninder Singh, Kush Varshney, Laura Onasch-Vera, Amulya Yadav, Tom Yan, and Xinran Yuan. I would also like to thank the department staff, especially Diane Stidle, for their very prompt help whenever anything was needed with respect to or around the department.

A special thanks to all my friends who helped make my time at CMU a wonderful and unforgettable period of my life: especially Ian Char, Renato Negrinho, Biswajit Paria, and Tom Yan, but also Young Chung, Jeremy Cohen, Mandy Coston, Ben Cowley, Nic Dalmasso, Chris Dann, Deepak Dilipkumar, Avi Dubey, Karan Goel, Audrey Huang, Conor Igoe, Anson Kahng, Lisa Lee, Jeff Li, Terrance Liu, Octavio Mesner, Vaishnavh Nagarajan, Willie Neiswanger, Tolani Olarinre, Ben Parr, Darshan Patil, Barun Patra, Anthony Platanios, Adarsh Prasad, Ariel Procaccia (yes, again :D), Paul Schydlo, Otilia

Stretcu, Mariya Toneva, Jake Tyo, Ezra Winston, Han Zhao, Xun Zheng, and many more I may have missed. I will never forget the fun times we had in the MLD PhD Lounge, especially the countless hours of playing smash, the board game nights, random chats, rock climbing, MLD retreats, working out, hanging out outside work and so on. I am especially extremely grateful to all of those who have been extremely supportive of me when I came out to them for the first time, and their support has definitely been priceless. These friends helped me pursue several non-academic trajectories of my life, and I cannot thank them enough for that.

Lastly, it goes without saying, I would not have even been in a place to be able to start my PhD if my parents had not given me the opportunities I had growing up. In closing, I would again like to thank everyone who has been very supportive of me, and helped keep my positive energy burning :)

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Multiagent preferences are the basic ingredient in some of the most well-studied areas of algorithmic game theory — computational social choice and fair division. In classical social choice and voting, there are n agents and m candidates, and each of these agents has a preference ordering or ranking over these m alternatives. Given the rankings of all the n agents, the goal is to find a winning candidate (or a consensus ranking) that is the most "fair" outcome. In this thesis, we look into several variants of this standard setting. For instance, we consider when there are an uncountably infinite number of alternatives, but each of them is defined by a set of d features. In such a setting, we cannot elucidate the entire ranking over the alternatives for any given voter, and hence need to learn this ranking by collecting a few pairwise comparisons from them, followed by generalizing to the rest of the alternative space. We also look into whether it is meaningful to pool all the comparisons together and learn a single community wide ranking directly, instead of learning individual rankings for the voters, followed by aggregating them.

Other variants we consider are settings where agents' preferences are very different from the realm of rankings over candidates. For instance, we consider the setting of a markov decision process, where we have multiple agents, but each of them with a different reward function (representing their preferences for this problem). Our goal is then to find a single policy that everyone would be "happy" with. Yet another example is the setting of a conference peer review system, where the agents are the reviewers of the conference, and their preferences are given by the defining characteristics they use to choose which papers are to be accepted at the conference. Finally, we also consider the setting where agents' preferences are defined by utility functions over a given set of outcomes, and our goal is to learn a classifier that is fair with respect to these preferences. Broadly speaking, this thesis tackles problems in three areas: (i) fairness in machine learning, (ii) voting and social choice, and (iii) reinforcement learning, each of them handling multiagent preferences with machine learning.

Fairness in Machine Learning. Machine learning models can inherit biases from the data they have been trained on. These biases in the data could arise because of a sample bias in how the data was collected, discrimination that occured in the past or because of

unconscious human bias. To handle these cases, machine learning is performed with certain fairness constraints imposed to make it "fair." The two most common notions of fairness are individual and group fairness. Group fairness is quite practical, but only guarantees very weak properties. On the other hand, individual fairness offers strong guarantees but is generally difficult to operationalize. This is because its most common form requires the existence and knowledge of a "magical" task-specific distance function. Hence, we propose a new measure of individual fairness, called *envy-freeness*. Envy-freeness is a well-known property in the fair division literature, but we argue is a compelling notion of fairness for classification tasks as well, especially when individuals have very heterogeneous preferences. Our technical focus is the *generalizability* of envy-free classification, i.e., we study the conditions under which a classifier that is envy free on a sample would be almost envy free with respect to the underlying distribution with high probability.

Voting and Social Choice. In this part, we study four problems, each at the intersection of machine learning and social choice. For the first, consider a setting with repeated aggregation of objective opinions. For example, suppose a group of engineers are trying to decide which prototype to develop, based on an objective measure of success such as projected market share. Standard voting systems treat all voters equally. We argue that perhaps this can be improved: Voters who have supported good choices in the past, and hence objectively have more expertise on the matter, should be given higher weight than voters who have supported bad ones – leading to better choices overall. To be able to design such weighting schemes, we draw on *no-regret learning*.

Second, we present a general approach to automating societal decisions, drawing on machine learning and computational social choice. The aim is to be able to predict what society would have chosen given a particular delimma at hand. In such a setting, the space of alternatives may be uncountably infinite, each alternative defined by a set of d features. The learning to rank literature studies algorithms to predict the ranking over the whole space of alternatives, given only a few comparisons between them. In our setting, this is additionally coupled with the fact that we have multiple voters, each with their own preferences. Our general approach to solving this problem, which we term virtual democracy, involves four steps: (i) collecting pairwise comparisons from voters, (ii) learning a model of preferences for each voter, (iii) summarizing all the models into a concise summary model and finally (iv) aggregating these preferences to make a decision. We provide a concrete algorithm that instantiates our approach; some of its crucial steps are informed by a new theory of *swap-dominance efficient* voting rules.

Third, we aim to improve the conference peer review system by tackling one of the subjective aspect of reviewing: mapping of criteria scores of papers to final recommendations. It is common to see a handful of reviewers reject a highly novel paper, because they view, say, extensive experiments as far more important than novelty, whereas the community as a whole would have embraced the paper. We define this mapping of criteria scores to final recommendations as the preference of the reviewer, and aim to find a consensus decision for the entire conference. In this work, we present a framework — based on $L(p, q)$-norm empirical risk minimization — for learning the community's aggregate mapping. And we

draw on computational social choice to identify desirable values of p and q.

Finally, we study axioms satisfied by learning from pairwise comparisons, and, its implications when these comparisons are obtained by pooling them from multiple agents. Our virtual demoncracy framework to automating societal decisions involved four steps. Crucially, the learning step (step ii) can be sample inefficient, as we are learning a different model for each voter independently. In particular, with only a few comparisons to learn from, we could either succumb to overfitting, or be forced to use a model with very low model complexity. This can be avoided by pooling all the pairwise comparisons together, leading to a much larger dataset, and directly learning an *aggregate* model from this pooled dataset. This work studies whether this is meaningful even from a social choice perspective. In particular, we show that for a large class of random utility models, the MLE satisfies a Pareto efficiency condition and a strong monotonicity property. While on the other hand, these models fail certain other consistency conditions like pairwise majority consistency and separability.

Reinforcement Learning. In this part, we study three problems dealing with multiagent preferences arising in reinforcement learning settings. First, we study the problem of influence maximization in social networks in the presence of *contingencies*. Specifically, we focus on using influence maximization in public health domains for assisting low-resource communities, like spreading HIV awareness among homeless youth networks, where *contingencies* are common. It is very difficult in these domains to ensure that the seed nodes are influenced, as these nodes correspond to homeless youth, and influencing them requires contacting them, followed by convincing them to attend training sessions. Unfortunately, previous work on influence maximization assumes that chosen influencers can be influenced with certainty, and hence are fairly suboptimal in this setting. In this work, we propose the *Contingency Aware Influence Maximization* problem to model this problem, cast it as a partially observable markov decision process, and propose a custom POMDP planner to solve it.

Second, we look into the problem of incorporating morality into reinforcement learning agents. To ensure that these agents behave in ways aligned with the values of society, we need to develop techniques that allow these agents to maximize their reward in the environment, while at the same time following these implicit constraints of society. In cases where we have access to the exact list of all the possible rules to follow, or a reward function of said morality, one could enforce these as constraints on the agent, or couple it with the reward function. But, in general, it may be extremely difficult to list them exactly and in its entirely. Hence, in this work we learn these constraints from moral demonstrations of the task (which may be imperfect from the standpoint of optimizing environmental rewards) via inverse reinforcement learning. Then, a contextual bandit-based orchestrator chooses between the learned moral policy and the environment reward maximizing policy at each point of time, depending on the context of the state. This additionally provides us with a layer of transparency, as we know which policy is being played at each point of time.

Finally, we consider a variant of this setting, where we have a markov decision process, and multiple agents, each with a different reward function. This reward function can be

viewed as the preference of the agent. Unlike previous work mentioned above, instead of picking between the different optimal policies (albeit, based on the context at each point of time), our goal is to learn a more truly "aggregate" policy that aligns with these preferences. In this work, we assume that the agents are like-minded, in the sense that their reward functions could be seen as random perturbations of an underlying reward function. Under this assumption, we demonstrate that by pooling the optimal trajectories from all the agents, followed by applying inverse reinforcement learning algorithms satisfying certain properties, we are able to learn a policy that is approximately optimal, and, no algorithm can have a better performance than this in the worst case. Next, we study the same problem but in the bandit setting – that is, we have access to the optimal arms of all the agents, and need to learn the optimal arm w.r.t. the underlying reward function – we term this the inverse bandit problem.

1.1 Overview of Thesis Contributions

This section presents an overview of each of the chapters of the thesis.

1.1.1 Fairness in Machine Learning

Chapter 2: Envy-Free Classification. In classic fair division problems such as cake cutting and rent division, *envy-freeness* requires that each individual (weakly) prefer his allocation to anyone else's. In this chapter, we argue that envy-freeness also provides a compelling notion of fairness for classification tasks, especially when individuals have heterogeneous preferences. Our technical focus is the *generalizability* of envy-free classification, i.e., understanding whether a classifier that is envy free on a sample would be almost envy free with respect to the underlying distribution with high probability. Our main result establishes that a small sample is sufficient to achieve such guarantees, when the classifier in question is a mixture of deterministic classifiers that belong to a family of low Natarajan dimension. We also design and implement an algorithm that learns such envy-free classifiers on the sample.

1.1.2 Voting and Social Choice

Chapter 3: Weighted Voting Via No-Regret Learning. Voting systems typically treat all voters equally. In this chapter, we argue that perhaps they should not: Voters who have supported good choices in the past should be given higher weight than voters who have supported bad ones. To develop a formal framework for desirable weighting schemes, we draw on *no-regret learning*. Specifically, given a voting rule, we wish to design a weighting scheme such that applying the voting rule, with voters weighted by the scheme, leads to choices that are almost as good as those endorsed by the best voter in hindsight. We derive possibility and impossibility results for the existence of such weighting schemes, depending on whether the voting rule and the weighting scheme are deterministic or randomized, as well as on the social choice axioms satisfied by the voting rule.

Chapter 4: Virtual Democracy: A Voting-Based Framework for Automating Decisions. In this chapter, we present a general approach to automating decisions, drawing on machine learning and computational social choice. In a nutshell, we propose to *learn* a model of societal preferences, and, when faced with a specific dilemma at runtime, efficiently *aggregate* those preferences to identify a desirable choice. We provide a concrete algorithm that instantiates our approach; some of its crucial steps are informed by a new theory of *swap-dominance efficient* voting rules. Finally, as a proof of concept, we implement and evaluate a system for decision making in the autonomous vehicle domain, using preference data collected from 1.3 million people through the Moral Machine website.

Chapter 5: Loss Functions, Axioms, and Peer Review. It is common to see a handful of reviewers reject a highly novel paper, because they view, say, extensive experiments as far more important than novelty, whereas the community as a whole would have embraced the paper. More generally, the disparate mapping of criteria scores to final recommendations by different reviewers is a major source of inconsistency in peer review. In this chapter, we present a framework inspired by empirical risk minimization (ERM) for learning the community's aggregate mapping. The key challenge that arises is the specification of a loss function for ERM. We consider the class of $L(p, q)$ loss functions, which is a matrix-extension of the standard class of L_p losses on vectors; here the choice of the loss function amounts to choosing the hyperparameters $p, q \in [1, \infty]$. To deal with the absence of ground truth in our problem, we instead draw on computational social choice to identify desirable values of the hyperparameters p and q. Specifically, we characterize $p = q = 1$ as the only choice of these hyperparameters that satisfies three natural axiomatic properties. Finally, we implement and apply our approach to reviews from IJCAI 2017.

Chapter 6: Axioms for Learning from Pairwise Comparisons. To be well-behaved, systems that process preference data must satisfy certain conditions identified by economic decision theory and by social choice theory. In ML, preferences and rankings are commonly learned by fitting a probabilistic model to noisy preference data. The behavior of this learning process from the view of economic theory has previously been studied for the case where the data consists of rankings. In practice, it is more common to have only pairwise comparison data, and the formal properties of the associated learning problem are more challenging to analyze. In this chapter, we show that a large class of random utility models (including the Thurstone–Mosteller Model), when estimated using the MLE, satisfy a Pareto efficiency condition. These models also satisfy a strong monotonicity property, which implies that the learning process is responsive to input data. On the other hand, we show that these models fail certain other consistency conditions from social choice theory, and in particular do not always follow the majority opinion. Our results inform existing and future applications of random utility models for societal decision making.

1.1.3 Reinforcement Learning

Chapter 7: Please be an Influencer? Contingency-Aware Influence Maximization. Most previous work on influence maximization in social networks assumes that the chosen influencers (or *seed nodes*) can be influenced with certainty (i.e., with no contingencies). In this chapter, we focus on using influence maximization in public health domains for assisting low-resource communities, where *contingencies* are common. It is very difficult in these domains to ensure that the seed nodes are influenced, as influencing them entails contacting/convincing them to attend training sessions, which may not always be possible. Unfortunately, previous state-of-the-art algorithms for influence maximization are unusable in this setting. This chapter tackles this challenge via the following four contributions: (i) we propose the Contingency Aware Influence Maximization problem and analyze it theoretically; (ii) we cast this problem as a Partially Observable Markov Decision Process and propose CAIMS (a novel POMDP planner) to solve it, which leverages a natural action space factorization associated with real-world social networks; and (iii) we provide extensive simulation results to compare CAIMS with existing state-of-the-art influence maximization algorithms. Finally, (iv) we provide results from a real-world feasibility trial conducted to evaluate CAIMS, in which key influencers in homeless youth social networks were influenced in order to spread awareness about HIV.

Chapter 8: Teaching AI Agents Ethical Values Using Reinforcement Learning and Policy Orchestration. Autonomous cyber-physical agents play an increasingly large role in our lives. To ensure that they behave in ways aligned with the values of society, we must develop techniques that allow these agents to not only maximize their reward in an environment, but also to learn and follow the implicit constraints of society. In this chapter, we detail a novel approach that uses inverse reinforcement learning to learn a set of unspecified constraints from demonstrations and reinforcement learning to learn to maximize environmental rewards. A contextual bandit-based orchestrator then picks between the two policies: constraint-based and environment reward-based. The contextual bandit orchestrator allows the agent to mix policies in novel ways, taking the best actions from either a reward-maximizing or constrained policy. In addition, the orchestrator is transparent on which policy is being employed at each time step. We test our algorithms using Pac-Man and show that the agent is able to learn to act optimally, act within the demonstrated constraints, and mix these two functions in complex ways.

Chapter 9: Inverse Reinforcement Learning from Like-Minded Teachers. In this chapter, we study the problem of learning a policy in a Markov decision process (MDP) based on observations of the actions taken by multiple teachers. We assume that the teachers are like-minded in that their reward functions — while different from each other — are random perturbations of an underlying reward function. Under this assumption, we demonstrate that inverse reinforcement learning algorithms that satisfy a certain property — that of *matching feature expectations* — yield policies that are approximately optimal with respect to the underlying reward function, and that no algorithm can do better in the worst case. We also show how to efficiently recover the optimal policy when

the MDP has one state — a setting that is akin to multi-armed bandits. Finally, we support this with experiments on non-trivial bandit problems, with varying parameters.

1.2 Bibliographic Remarks

The research presented in this thesis is based on joint work with several co-authors, listed below. And this thesis only includes work where the author was the, or one of the, primary contributors. Chapter 2 is based on joint work with Nina Balcan, Travis Dick and Ariel Procaccia [Bal+19b]. Chapter 3 is based on joint work with Nika Haghtalab and Ariel Procaccia [HNP18]. Chapter 4 is based on joint work with Neil Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar and Ariel Procaccia [Noo+18]. Chapter 5 is based on joint work with Nihar Shah and Ariel Procaccia [NSP20]. Chapter 6 is based on joint work with Dominik Peters and Ariel Procaccia [NPP20]. Chapter 7 is based on joint work with Amulya Yadav, Eric Rice, Laura Onasch-Vera, Leandro S. Marcolino and Milind Tambe [Yad+18]. Chapter 8 is based on joint work with Djallel Bouneffouf, Nicholas Mattei, Rachita Chandra, Piyush Madan, Kush Varshney, Murray Campbell, Moninder Singh and Francesca Rossi [Noo+19]. And finally, Chapter 9 is based on joint work with Tom Yan and Ariel Procaccia [NYP20].

1.2.1 Non-thesis research

I have also contributed to a few other papers during my Ph.D. studies: [Kah+19] and [Lee+19], but they are not included in my thesis.

Part I

Fairness in Machine Learning

Chapter 2

Envy-Free Classification

In classic fair division problems such as cake cutting and rent division, *envy-freeness* requires that each individual (weakly) prefer his allocation to anyone else's. On a conceptual level, we argue that envy-freeness also provides a compelling notion of fairness for classification tasks, especially when individuals have heterogeneous preferences. Our technical focus is the *generalizability* of envy-free classification, i.e., understanding whether a classifier that is envy free on a sample would be almost envy free with respect to the underlying distribution with high probability. Our main result establishes that a small sample is sufficient to achieve such guarantees, when the classifier in question is a mixture of deterministic classifiers that belong to a family of low Natarajan dimension.

2.1 Introduction

The study of fairness in machine learning is driven by an abundance of examples where learning algorithms were perceived as discriminating against protected groups [Swe13; DTD15]. Addressing this problem requires a conceptual — perhaps even philosophical — understanding of what fairness means in this context. In other words, the million dollar question is (arguably[1]) this: What are the formal constraints that fairness imposes on learning algorithms?

In this paper, we propose a new measure of algorithmic fairness. It draws on an extensive body of work on rigorous approaches to fairness, which — modulo one possible exception (see Section 2.1.2) — has not been tapped by machine learning researchers: the literature on *fair division* [BT96b; Mou03]. The most prominent notion is that of *envy-freeness* [Fol67; Var74], which, in the context of the allocation of goods, requires that the utility of each individual for his allocation be at least as high as his utility for the allocation of any other individual; for six decades, it has been the gold standard of fairness for problems such as cake cutting [RW98; Pro13] and rent division [Su99; Gal+17]. In the classification setting, envy-freeness would simply mean that the utility of each individual for his distribution over outcomes is at least as high as his utility for the distribution over outcomes assigned

[1]Certain papers take a somewhat different view [Kil+17].

to any other individual.

It is important to say upfront that envy-freeness is *not* suitable for several widely-studied problems where there are only two possible outcomes, one of which is 'good' and the other 'bad'; examples include predicting whether an individual would default on a loan, and whether an offender would recidivate. In these degenerate cases, envy-freeness would require that the classifier assign each and every individual the exact same probability of obtaining the 'good' outcome, which, clearly, is not a reasonable constraint.

By contrast, we are interested in situations where there is a diverse set of possible outcomes, and individuals have diverse preferences for those outcomes. For example, consider a system responsible for displaying credit card advertisements to individuals. There are many credit cards with different eligibility requirements, annual rates, and reward programs. An individual's utility for seeing a card's advertisement will depend on his eligibility, his benefit from the rewards programs, and potentially other factors. It may well be the case that an envy-free advertisement assignment shows Bob advertisements for a card with worse annual rates than those shown to Alice; this outcome is not unfair if Bob is genuinely more interested in the card offered to him. Such rich utility functions are also evident in the context of job advertisements [DTD15]: people generally want higher paying jobs, but would presumably have higher utility for seeing advertisements for jobs that better fit their qualifications and interests.

A second appealing property of envy-freeness is that its fairness guarantee binds at the level of individuals. Fairness notions can be coarsely characterized as being either individual notions, or group notions, depending on whether they provide guarantees to specific individuals, or only on average to a protected subgroup. The majority of work on fairness in machine learning focuses on group fairness [LRT11; Dwo+12; Zem+13; HPS16; Jos+16; Zaf+17].

There is, however, one well-known example of individual fairness: the influential fair classification model of Dwork, Hardt, Pitassi, Reingold, and Zemel [Dwo+12]. The model involves a set of individuals and a set of outcomes. The centerpiece of the model is a *similarity metric* on the space of individuals; it is specific to the classification task at hand, and ideally captures the ethical ground truth about relevant attributes. For example, a man and a woman who are similar in every other way should be considered similar for the purpose of credit card offerings, but perhaps not for lingerie advertisements. Assuming such a metric is available, fairness can be naturally formalized as a Lipschitz constraint, which requires that individuals who are close according to the similarity metric be mapped to distributions over outcomes that are close according to some standard metric (such as total variation).

As attractive as this model is, it has one clear weakness from a practical viewpoint: the availability of a similarity metric. Dwork, Hardt, Pitassi, Reingold, and Zemel [Dwo+12] are well aware of this issue; they write that justifying this assumption is "one of the most challenging aspects" of their approach. They add that "in reality the metric used will most likely only be society's current best approximation to the truth." But, despite recent progress on automating ethical decisions in certain domains [Noo+18; Fre+20], the task-specific nature of the similarity metric makes even a credible approximation thereof seem unrealistic. In particular, if one wanted to learn a similarity metric, it is unclear what type

of examples a relevant dataset would consist of.

In place of a metric, envy-freeness requires access to individuals' utility functions, but — by contrast — we do not view this assumption as a barrier to implementation. Indeed, there are a variety of techniques for learning utility functions [CKO01; NJ04; Bal+12]. Moreover, in our running example of advertising, one can use standard measures like expected click-through rate (CTR) as a good proxy for utility.

It is worth noting that the classification setting is different from classic fair division problems in that the "goods" (outcomes) are non-excludable. In fact, one envy-free solution simply assigns each individual to his favorite outcome. But this solution may be severely suboptimal according to another (standard) component of our setting, the *loss function*, which, in the examples above, might represent the expected revenue from showing an ad to an individual. Typically the loss function is not perfectly aligned with individual utilities, and, therefore, it may be possible to achieve smaller loss than the naïve solution without violating the envy-freeness constraint.

In summary, we view envy-freeness as a compelling, well-established, and, importantly, practicable notion of individual fairness for classification tasks with a diverse set of outcomes when individuals have heterogeneous preferences. Our goal is to understand its learning-theoretic properties.

2.1.1 Our Results

The challenge is that the space of individuals is potentially huge, yet we seek to provide universal envy-freeness guarantees. To this end, we are given a sample consisting of individuals drawn from an unknown distribution. We are interested in learning algorithms that minimize loss, subject to satisfying the envy-freeness constraint, *on the sample*. Our primary technical question is that of generalizability, that is, *given a classifier that is envy free on a sample, is it approximately envy free on the underlying distribution?* Surprisingly, Dwork, Hardt, Pitassi, Reingold, and Zemel [Dwo+12] do not study generalizability in their model, and we are aware of only one subsequent paper that takes a learning-theoretic viewpoint on individual fairness and gives theoretical guarantees (see Section 2.1.2).

In Section 2.3, we do not constrain the classifier. Therefore, we need some strategy to extend a classifier that is defined on a sample; assigning an individual the same outcome as his *nearest neighbor* in the sample is a popular choice. However, we show that *any* strategy for extending a classifier from a sample, on which it is envy free, to the entire set of individuals is unlikely to be approximately envy free on the distribution, unless the sample is exponentially large.

For this reason, in Section 2.4, we focus on structured families of classifiers. On a high level, our goal is to relate the combinatorial richness of the family to generalization guarantees. One obstacle is that standard notions of dimension do not extend to the analysis of randomized classifiers, whose range is *distributions* over outcomes (equivalently, real vectors). We circumvent this obstacle by considering mixtures of *deterministic* classifiers that belong to a family of bounded Natarajan dimension (an extension of the well-known VC dimension to multi-class classification). Our main theoretical result asserts that, under this assumption, envy-freeness on a sample does generalize to the underlying distribution,

even if the sample is relatively small (its size grows almost linearly in the Natarajan dimension).

Finally, in Section 2.5, we design and implement an algorithm that learns (almost) envy-free mixtures of linear one-vs-all classifiers. We present empirical results that validate our computational approach, and indicate good generalization properties even when the sample size is small.

2.1.2 Related Work

Conceptually, our work is most closely related to work by Zafar, Valera, Gomez-Rodriguez, Gummadi, and Weller [Zaf+17]. They are interested in group notions of fairness, and advocate preference-based notions instead of parity-based notions. In particular, they assume that each group has a utility function for *classifiers*, and define the *preferred treatment* property, which requires that the utility of each group for its own classifier be at least its utility for the classifier assigned to any other group. Their model and results focus on the case of binary classification where there is a desirable outcome and an undesirable outcome, so the utility of a group for a classifier is simply the fraction of its members that are mapped to the desirable outcome. Although, at first glance, this notion seems similar to envy-freeness, it is actually fundamentally different.[2] Our paper is also completely different from that of Zafar, Valera, Gomez-Rodriguez, Gummadi, and Weller in terms of technical results; theirs are purely empirical in nature, and focus on the increase in accuracy obtained when parity-based notions of fairness are replaced with preference-based ones.

Concurrent work by Rothblum and Yona [RY18] provides generalization guarantees for the metric notion of individual fairness introduced by Dwork, Hardt, Pitassi, Reingold, and Zemel [Dwo+12], or, more precisely, for an approximate version thereof. There are two main differences compared to our work: first, we propose envy-freeness as an alternative notion of fairness that circumvents the need for a similarity metric. Second, they focus on randomized *binary* classification, which amounts to learning a real-valued function, and so are able to make use of standard Rademacher complexity results to show generalization. By contrast, standard tools do not directly apply in our setting. It is worth noting that several other papers provide generalization guarantees for notions of group fairness, but these are more distantly related to our work [Zem+13; Woo+17; Don+18; Kea+18; Hb+18].

2.2 The Model

We assume that there is a space $\mathcal{X}$ of individuals, a finite space $\mathcal{Y}$ of outcomes, and a utility function $u : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ encoding the preferences of each individual for the outcomes in $\mathcal{Y}$. In the advertising example, individuals are users, outcomes are advertisements, and the utility function reflects the benefit an individual derives from being shown a particular

[2] On a philosophical level, the fair division literature deals exclusively with individual notions of fairness. In fact, even in group-based extensions of envy-freeness [MS17] the allocation is shared by groups, but individuals must not be envious. We subscribe to the view that group-oriented notions (such as statistical parity) are objectionable, because the outcome can be patently unfair to individuals.

advertisement. For any distribution $p \in \Delta(\mathcal{Y})$ (where $\Delta(\mathcal{Y})$ is the set of distributions over $\mathcal{Y}$) we let $u(x, p) = \mathbb{E}_{y\sim p}[u(x, y)]$ denote individual x's expected utility for an outcome sampled from p. We refer to a function $h : \mathcal{X} \to \Delta(\mathcal{Y})$ as a *classifier*, even though it can return a distribution over outcomes.

2.2.1 Envy-Freeness

Roughly speaking, a classifier $h : \mathcal{X} \to \Delta(\mathcal{Y})$ is envy free if no individual prefers the outcome distribution of someone else over his own.

Definition 2.2.1. A classifier $h : \mathcal{X} \to \Delta(\mathcal{Y})$ is *envy free (EF)* on a set S of individuals if $u(x, h(x)) \geq u(x, h(x'))$ for all $x, x' \in S$. Similarly, h is (α, β)-*EF* with respect to a distribution P on $\mathcal{X}$ if

$$\Pr_{x,x'\sim P}\big(u(x, h(x)) < u(x, h(x')) - \beta\big) \leq \alpha.$$

Finally, h is (α, β)-*pairwise EF* on a set of pairs of individuals $S = \{(x_i, x_i')\}_{i=1}^n$ if

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{u(x_i, h(x_i)) < u(x_i, h(x_i')) - \beta\} \leq \alpha.$$

Any classifier that is EF on a sample S of individuals is also (α, β)-pairwise EF on any pairing of the individuals in S, for any $\alpha \geq 0$ and $\beta \geq 0$. The weaker pairwise EF condition is all that is required for our generalization guarantees to hold.

2.2.2 Optimization and Learning

Our formal learning problem can be stated as follows. Given sample access to an unknown distribution P over individuals $\mathcal{X}$ and their utility functions, and a known loss function $\ell : \mathcal{X} \times \mathcal{Y} \to [0, 1]$, find a classifier $h : \mathcal{X} \to \Delta(\mathcal{Y})$ that is (α, β)-EF with respect to P minimizing expected loss $\mathbb{E}_{x\sim P}[\ell(x, h(x))]$, where for $x \in \mathcal{X}$ and $p \in \Delta(\mathcal{Y})$, $\ell(x, p) = \mathbb{E}_{y\sim p}[\ell(x, y)]$.

We follow the empirical risk minimization (ERM) learning approach, i.e., we collect a sample of individuals drawn i.i.d from P and find an EF classifier with low loss on the sample. Formally, given a sample of individuals $S = \{x_1, \ldots, x_n\}$ and their utility functions $u_{x_i}(\cdot) = u(x_i, \cdot)$, we are interested in a classifier $h : S \to \Delta(\mathcal{Y})$ that minimizes $\sum_{i=1}^n \ell(x_i, h(x_i))$ among all classifiers that are EF on S.

Recall that we consider randomized classifiers that can assign a distribution over outcomes to each of the individuals. However, one might wonder whether the EF classifier that minimizes loss on a sample happens to always be deterministic. Or, at least, the optimal deterministic classifier on the sample might incur a loss that is very close to that of the optimal randomized classifier. If this were true, we could restrict ourselves to classifiers of the form $h : \mathcal{X} \to \mathcal{Y}$, which would be much easier to analyze. Unfortunately, it turns out that this is not the case. In fact, there could be an arbitrary (multiplicative) gap between

the optimal randomized EF classifier and the optimal deterministic EF classifier. The intuition behind this is as follows. A deterministic classifier that has very low loss on the sample, but is not EF, would be completely discarded in the deterministic setting. On the other hand, a randomized classifier could take this loss-minimizing deterministic classifier and mix it with a classifier with high "negative envy", so that the mixture ends up being EF and at the same time has low loss. This is made concrete in the following example.

Example 2.2.2. Let $S = \{x_1, x_2\}$ and $\mathcal{Y} = \{y_1, y_2, y_3\}$. Let the loss function be such that

$$\begin{array}{lll} \ell(x_1, y_1) = 0 & \ell(x_1, y_2) = 1 & \ell(x_1, y_3) = 1 \\ \ell(x_2, y_1) = 1 & \ell(x_2, y_2) = 1 & \ell(x_2, y_3) = 0 \end{array}$$

Moreover, let the utility function be such that

$$\begin{array}{lll} u(x_1, y_1) = 0 & u(x_1, y_2) = 1 & u(x_1, y_3) = \dfrac{1}{\gamma} \\ u(x_2, y_1) = 0 & u(x_2, y_2) = 0 & u(x_2, y_3) = 1 \end{array}$$

where $\gamma > 1$. The only deterministic classifier with a loss of 0 is h_0 such that $h_0(x_1) = y_1$ and $h_0(x_2) = y_3$. But, this is not EF, since $u(x_1, y_1) < u(x_1, y_3)$. Furthermore, every other deterministic classifier has a total loss of at least 1, causing the optimal deterministic EF classifier to have loss of at least 1.

To show that randomized classifiers can do much better, consider the randomized classifier h_* such that $h_*(x_1) = (1 - 1/\gamma, 1/\gamma, 0)$ and $h_*(x_2) = (0, 0, 1)$. This classifier can be seen as a mixture of the classifier h_0 of 0 loss, and the deterministic classifier h_e, where $h_e(x_1) = y_2$ and $h_e(x_2) = y_3$, which has high "negative envy". One can observe that this classifier h_* is EF, and has a loss of just $1/\gamma$. Hence, the loss of the optimal randomized EF classifier is γ times smaller than the loss of the optimal deterministic one, for any $\gamma > 1$.

2.3 Arbitrary Classifiers

An important (and typical) aspect of our learning problem is that the classifier h needs to provide an outcome distribution for every individual, not just those in the sample. For example, if h chooses advertisements for visitors of a website, the classifier should still apply when a new visitor arrives. Moreover, when we use the classifier for new individuals, it must continue to be EF. In this section, we consider two-stage approaches that first choose outcome distributions for the individuals in the sample, and then extend those decisions to the rest of $\mathcal{X}$.

In more detail, we are given a sample $S = \{x_1, \ldots, x_n\}$ of individuals and a classifier $h : S \to \Delta(\mathcal{Y})$ assigning outcome distributions to each individual. Our goal is to extend these assignments to a classifier $\overline{h} : \mathcal{X} \to \Delta(\mathcal{Y})$ that can be applied to new individuals as well. For example, h could be the loss-minimizing EF classifier on the sample S.

For this section, we assume that $\mathcal{X}$ is equipped with a distance metric d. Moreover, we assume in this section that the utility function u is L-Lipschitz on $\mathcal{X}$. That is, for every $y \in \mathcal{Y}$ and for all $x, x' \in \mathcal{X}$, we have $|u(x, y) - u(x', y)| \leq L \cdot d(x, x')$.

Under the foregoing assumptions, one natural way to extend the classifier on the sample to all of $\mathcal{X}$ is to assign new individuals the same outcome distribution as their nearest neighbor in the sample. Formally, for a set $S \subset \mathcal{X}$ and any individual $x \in \mathcal{X}$, let $\mathrm{NN}_S(x) \in \arg\min_{x' \in S} d(x, x')$ denote the nearest neighbor of x in S with respect to the metric d (breaking ties arbitrarily). The following simple result (whose proof is relegated to Appendix A.2) establishes that this approach preserves envy-freeness in cases where the sample is exponentially large.

Theorem 2.3.1. *Let d be a metric on $\mathcal{X}$, P be a distribution on $\mathcal{X}$, and u be an L-Lipschitz utility function. Let S be a set of individuals such that there exists $\hat{\mathcal{X}} \subset \mathcal{X}$ with $P(\hat{\mathcal{X}}) \geq 1 - \alpha$ and $\sup_{x \in \hat{\mathcal{X}}} d(x, \mathrm{NN}_S(x)) \leq \beta/(2L)$. Then for any classifier $h : S \to \Delta(\mathcal{Y})$ that is EF on S, the extension $\overline{h} : \mathcal{X} \to \Delta(\mathcal{Y})$ given by $\overline{h}(x) = h(\mathrm{NN}_S(x))$ is (α, β)-EF on P.*

The conditions of Theorem 2.3.1 require that the set of individuals S is a $\beta/(2L)$-net for at least a $(1 - \alpha)$-fraction of the mass of P on $\mathcal{X}$. In several natural situations, an exponentially large sample guarantees that this occurs with high probability. For example, if $\mathcal{X}$ is a subset of $\mathbb{R}^q$, $d(x, x') = \|x - x'\|_2$, and $\mathcal{X}$ has diameter at most D, then for any distribution P on $\mathcal{X}$, if S is an i.i.d. sample of size $O(\frac{1}{\alpha}(\frac{LD\sqrt{q}}{\beta})^q(q \log \frac{LD\sqrt{q}}{\beta} + \log \frac{1}{\delta}))$, it will satisfy the conditions of Theorem 2.3.1 with probability at least $1 - \delta$. This sampling result is folklore, but, for the sake of completeness, we prove it in Lemma A.2.1 of Appendix A.2.

However, the exponential upper bound given by the nearest neighbor strategy is as far as we can go in terms of generalizing envy-freeness from a sample (without further assumptions). Specifically, our next result establishes that *any* algorithm — even randomized — for extending classifiers from the sample to the entire space $\mathcal{X}$ requires an exponentially large sample of individuals to ensure envy-freeness on the distribution P. The proof of Theorem 2.3.2 can be found in Appendix A.2.

Theorem 2.3.2. *There exists a space of individuals $\mathcal{X} \subset \mathbb{R}^q$, and a distribution P over $\mathcal{X}$ such that, for every randomized algorithm $\mathcal{A}$ that extends classifiers on a sample to $\mathcal{X}$, there exists an L-Lipschitz utility function u such that, when a sample of individuals S of size $n = 4^q/2$ is drawn from P without replacement, there exists an EF classifier on S for which, with probability at least $1 - 2\exp(-4^q/100) - \exp(-4^q/200)$ jointly over the randomness of $\mathcal{A}$ and S, its extension by $\mathcal{A}$ is not (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$.*

We remark that a similar result would hold even if we sampled S with replacement; we sample here without replacement purely for ease of exposition.

2.4 Low-Complexity Families of Classifiers

In this section we show that (despite Theorem 2.3.2) generalization for envy-freeness is possible using much smaller samples of individuals, as long as we restrict ourselves to classifiers from a family of relatively low complexity.

In more detail, two classic complexity measures are the VC-dimension [VC71] for binary classifiers, and the Natarajan dimension [Nat89] for multi-class classifiers. However, to the best of our knowledge, there is no suitable dimension directly applicable to functions

ranging over distributions, which in our case can be seen as $|\mathcal{Y}|$-dimensional real vectors. One possibility would be to restrict ourselves to deterministic classifiers of the type $h : \mathcal{X} \to \mathcal{Y}$, but we have seen in Section 2.2 that envy-freeness is a very strong constraint on deterministic classifiers. Instead, we will consider a family $\mathcal{H}$ consisting of randomized mixtures of m deterministic classifiers belonging to a family $\mathcal{G} \subset \{g : \mathcal{X} \to \mathcal{Y}\}$ of low Natarajan dimension. This allows us to adapt Natarajan-dimension-based generalization results to our setting while still working with randomized classifiers. The definition and relevant properties of the Natarajan dimension are summarized in Appendix A.1.

Formally, let $\vec{g} = (g_1, \ldots, g_m) \in \mathcal{G}^m$ be a vector of m functions in $\mathcal{G}$ and $\eta \in \Delta_m$ be a distribution over $[m]$, where $\Delta_m = \{p \in \mathbb{R}^m : p_i \geq 0, \sum_i p_i = 1\}$ is the m-dimensional probability simplex. Then consider the function $h_{\vec{g},\eta} : \mathcal{X} \to \Delta(\mathcal{Y})$ with assignment probabilities given by $\Pr(h_{\vec{g},\eta}(x) = y) = \sum_{i=1}^{m} \mathbb{I}\{g_i(x) = y\}\eta_i$. Intuitively, for a given individual x, $h_{\vec{g},\eta}$ chooses one of the g_i randomly with probability η_i, and outputs $g_i(x)$. Let

$$\mathcal{H}(\mathcal{G}, m) = \{h_{\vec{g},\eta} : \mathcal{X} \to \Delta(\mathcal{Y}) : \vec{g} \in \mathcal{G}^m, \eta \in \Delta_m\}$$

be the family of classifiers that can be written this way. Our main technical result shows that envy-freeness generalizes for this class.

Theorem 2.4.1. *Suppose $\mathcal{G}$ is a family of deterministic classifiers of Natarajan dimension d, and let $\mathcal{H} = \mathcal{H}(\mathcal{G}, m)$ for $m \in \mathbb{N}$. For any distribution P over $\mathcal{X}$, $\gamma > 0$, and $\delta > 0$, if $S = \{(x_i, x_i')\}_{i=1}^{n}$ is an i.i.d. sample of pairs drawn from P of size*

$$n \geq O\left(\frac{1}{\gamma^2}\left(dm^2 \log \frac{dm|\mathcal{Y}| \log(m|\mathcal{Y}|/\gamma)}{\gamma} + \log \frac{1}{\gamma}\right)\right),$$

then with probability at least $1 - \delta$, every classifier $h \in \mathcal{H}$ that is (α, β)-pairwise-EF on S is also $(\alpha + 7\gamma, \beta + 4\gamma)$-EF on P.

The proof of Theorem 2.4.1 is relegated to Appendix A.3. In a nutshell, it consists of two steps. First, we show that envy-freeness generalizes for finite classes. Second, we show that $\mathcal{H}(\mathcal{G}, m)$ can be approximated by a finite subset.

We remark that the theorem is only effective insofar as families of classifiers of low Natarajan dimension are useful. Fortunately, several prominent families indeed have low Natarajan dimension [DSS12], including one vs. all, multiclass SVM, tree-based classifiers, and error correcting output codes.

2.5 Implementation and Empirical Validation

So far we have not directly addressed the problem of *computing* the loss-minimizing envy-free classifier from a given family on a given sample of individuals. We now turn to this problem. Our goal is not to provide an end-all solution, but rather to provide evidence that computation will not be a long-term obstacle to implementing our approach.

In more detail, our computational problem is to find the loss-minimizing classifier h from a given family of randomized classifiers $\mathcal{H}$ that is envy free on a given a sample of individuals $S = \{x_1, \ldots, x_n\}$. For this classifier h to generalize to the distribution P,

Theorem 2.4.1 suggests that the family $\mathcal{H}$ to use is of the form $\mathcal{H}(\mathcal{G}, m)$, where $\mathcal{G}$ is a family of deterministic classifiers of low Natarajan dimension.

In this section, we let $\mathcal{G}$ be the family of *linear one-vs-all classifiers*. In particular, denoting $\mathcal{X} \subset \mathbb{R}^q$, each $g \in \mathcal{G}$ is parameterized by $\vec{w} = (w_1, w_2, \ldots, w_{|\mathcal{Y}|}) \in \mathbb{R}^{|\mathcal{Y}| \times q}$, where $g(x) = \text{argmax}_{y \in \mathcal{Y}} \left(w_y^\top x\right)$. This class $\mathcal{G}$ has a Natarajan dimension of at most $q|\mathcal{Y}|$. The optimization problem to solve in this case is

$$\begin{aligned} \min_{\vec{g} \in \mathcal{G}^m, \eta \in \Delta_m} \quad & \sum_{i=1}^{n} \sum_{k=1}^{m} \eta_k L(x_i, g_k(x_i)) \\ \text{s.t.} \quad & \sum_{k=1}^{m} \eta_k u(x_i, g_k(x_i)) \geq \sum_{k=1}^{m} \eta_k u(x_i, g_k(x_j)) \quad \forall (i,j) \in [n]^2. \end{aligned} \tag{2.1}$$

2.5.1 Algorithm

Observe that optimization problem (2.1) is highly non-convex and non-differentiable as formulated, because of the argmax computed in each of the $g_k(x_i)$. Another challenge is the combinatorial nature of the problem, as we need to find m functions from $\mathcal{G}$ along with their mixing weights. In designing an algorithm, therefore, we employ several tricks of the trade to achieve tractability.

Learning the mixture components. We first assume predefined mixing weights $\tilde{\eta}$, and *iteratively* learn mixture components based on them. Specifically, let $g_1, g_2, \ldots g_{k-1}$ denote the classifiers learned so far. To compute the next component g_k, we solve the optimization problem (2.1) with these components already in place (and assuming no future ones). This induces the following optimization problem.

$$\begin{aligned} \min_{g_k \in \mathcal{G}} \quad & \sum_{i=1}^{n} L(x_i, g_k(x_i)) \\ \text{s.t.} \quad & USF_{ii}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_i)) \geq USF_{ij}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_j)) \quad \forall (i,j) \in [n]^2, \end{aligned} \tag{2.2}$$

where $USF_{ij}^{(k-1)}$ denotes the expected utility i has for j's assignments so far, i.e., $USF_{ij}^{(k-1)} = \sum_{c=1}^{k-1} \tilde{\eta}_c u(x_i, g_c(x_j))$.

Solving the optimization problem (2.2) is still non-trivial because it remains non-convex and non-differentiable. To resolve this, we first soften the constraints[3]. Writing out the optimization problem in the form equivalent to introducing slack variables, we obtain

$$\begin{aligned} \min_{g_k \in \mathcal{G}} \quad & \sum_{i=1}^{n} L(x_i, g_k(x_i)) \\ & + \lambda \sum_{i \neq j} \max\left(USF_{ij}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_j)) - USF_{ii}^{(k-1)} - \tilde{\eta}_k u(x_i, g_k(x_i)), 0\right), \end{aligned} \tag{2.3}$$

[3]This may lead to solutions that are not exactly EF on the sample. Nonetheless, Theorem 2.4.1 still guarantees that there should not be much additional envy on the testing data.

where λ is a parameter that defines the trade-off between loss and envy-freeness. This optimization problem is still highly non-convex as $g_k(x_i) = \text{argmax}_{y\in\mathcal{Y}} w_y^\top x_i$, where $\vec{w}$ denotes the parameters of g_k. To solve this, we perform a convex relaxation on several components of the objective using the fact that $w_{g_k(x_i)}^\top x_i \geq w_{y'}^\top x_i$ for any $y' \in \mathcal{Y}$. Specifically, we have

$$L(x_i, g_k(x_i)) \leq \max_{y\in\mathcal{Y}} \left\{ L(x_i, y) + w_y^\top x_i - w_{y_i}^\top x_i \right\},$$

$$-u(x_i, g_k(x_i)) \leq \max_{y\in\mathcal{Y}} \left\{ -u(x_i, y) + w_y^\top x_i - w_{b_i}^\top x_i \right\}, \text{ and}$$

$$u(x_i, g_k(x_j)) \leq \max_{y\in\mathcal{Y}} \left\{ u(x_i, y) + w_y^\top x_j - w_{s_i}^\top x_j \right\},$$

where $y_i = \text{argmin}_{y\in\mathcal{Y}} L(x_i, y)$, $s_i = \text{argmin}_{y\in\mathcal{Y}} u(x_i, y)$ and $b_i = \text{argmax}_{y\in\mathcal{Y}}\ u(x_i, y)$. While we provided the key steps here, complete details and the rationale behind these choices are given in Appendix A.4. On a very high-level, these are inspired by multi-class SVMs. Finally, plugging these relaxations into (2.3), we obtain the following convex optimization problem to compute each mixture component.

$$\min_{\vec{w}\in\mathbb{R}^{|\mathcal{Y}|\times q}} \quad \sum_{i=1}^{n} \max_{y\in\mathcal{Y}} \left\{ L(x_i, y) + w_y^\top x_i - w_{y_i}^\top x_i \right\} + \lambda \sum_{i\neq j} \max \Big(USF_{ij}^{(k-1)} \tag{2.4}$$

$$+\tilde{\eta}_k \max_{y\in\mathcal{Y}} \left\{ u(x_i, y) + w_y^\top x_j - w_{s_i}^\top x_j \right\} - USF_{ii}^{(k-1)} + \tilde{\eta}_k \max_{y\in\mathcal{Y}} \left\{ -u(x_i, y) + w_y^\top x_i - w_{b_i}^\top x_i \right\}, 0 \Big).$$

Learning the mixing weights. Once the mixture components $\vec{g}$ are learned (with respect to the predefined mixing weights $\tilde{\eta}$), we perform an additional round of optimization to learn the optimal weights η for them. This can be done via the following linear program

$$\begin{aligned} \min_{\eta\in\Delta_m, \xi\in\mathbb{R}_{\geq 0}^{n\times n}} \quad & \sum_{i=1}^{n}\sum_{k=1}^{m} \eta_k L(x_i, g_k(x_i)) + \lambda \sum_{i\neq j} \xi_{ij} \\ \text{s.t.} \quad & \sum_{k=1}^{m} \eta_k u(x_i, g_k(x_i)) \geq \sum_{k=1}^{m} \eta_k u(x_i, g_k(x_j)) - \xi_{ij} \quad \forall (i,j). \end{aligned} \tag{2.5}$$

2.5.2 Methodology

To validate our approach, we have implemented our algorithm. However, we cannot rely on standard datasets, as we need access to both the features and the utility functions of individuals. Hence, we rely on synthetic data. All our code is included as supplementary material. Our experiments are carried out on a desktop machine with 16GB memory and an Intel Xeon(R) CPU E5-1603 v3 @ 2.80GHz×4 processor. To solve convex optimization problems, we use CVXPY [DB16; Agr+18].

In our experiments, we cannot compute the optimal solution to the original optimization problem (2.1), and there are no existing methods we can use as benchmarks. Hence, we generate the dataset such that we know the optimal solution upfront.

Specifically, to generate the whole dataset (both training and test), we first generate random classifiers $\vec{g}^\star \in \mathcal{G}^m$ by sampling their parameters $\vec{w}_1, \ldots \vec{w}_m \sim \mathcal{N}(0,1)^{|\mathcal{Y}|\times q}$, and

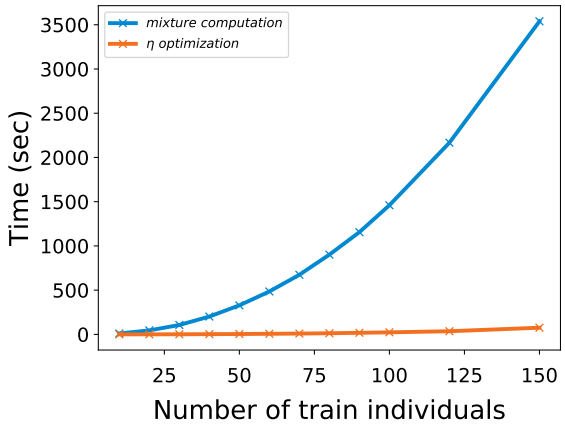


Figure 2.1: The algorithm's running time.

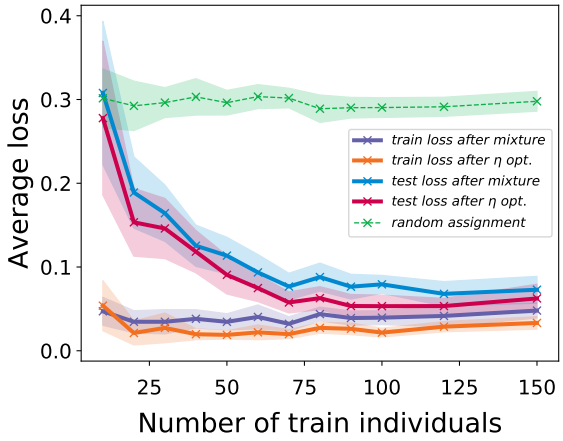


Figure 2.2: Training and test loss. Shaded error bands depict 95% confidence intervals.

generate $\eta^\star \in \Delta_m$ by drawing uniformly random weights in $[0, 1]$ and normalizing. We use $h_{\vec{g}^\star,\eta^\star}$ as the optimal solution of the dataset we generate. For each individual, we sample each feature value independently and u.a.r. in $[0, 1]$. For each individual x and outcome y, we set $L(x, y) = 0$ if $y \in \{g_k^\star(x) : k \in [m]\}$ and otherwise we sample $L(x, y)$ u.a.r. in $[0, 1]$. For the utility function u, we need to generate it such that the randomized classifier $h_{\vec{g}^\star,\eta^\star}$ is envy free on the dataset. For this, we set up a linear program and compute each of the values $u(x, y)$. Hence, $h_{\vec{g}^\star,\eta^\star}$ is envy free and has zero loss, so it is obviously the optimal solution. The dataset is split into 75% training data (to measure the accuracy of our solution to the optimization problem) and 25% test data (to evaluate generalizability).

For our experiments, we use the following parameters: $|\mathcal{Y}| = 10$, $q = 10$, $m = 5$, and $\lambda = 10.0$. We set the predefined weights to be $\tilde{\eta} = \left[\frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{2^{m-1}}, \frac{1}{2^{m-1}}\right]$.[4] In our experiments we vary the number of individuals, and each result is averaged over 25 runs. On each run, we generate a new ground-truth classifier $h_{\vec{g}^*,\eta^*}$, as well as new individuals, losses, and utilities.

2.5.3 Results

Figure 2.1 shows the time taken to compute the mixture components $\vec{g}$ and the optimal weights η, as the number of individuals in the training data increases. As we will see shortly, even though the η computation takes a very small fraction of the time, it can lead to non-negligible gains in terms of loss and envy.

Figure 2.2 shows the average loss attained on the training and test data by the algorithm immediately after computing the mixture components, and after the round of η optimization. It also shows the average loss attained (on both the training and test data)

[4]The reason for using an exponential decay is so that the subsequent classifiers learned are different from the previous ones. Using smaller weights might cause consecutive classifiers to be identical, thereby 'wasting' some of the components.

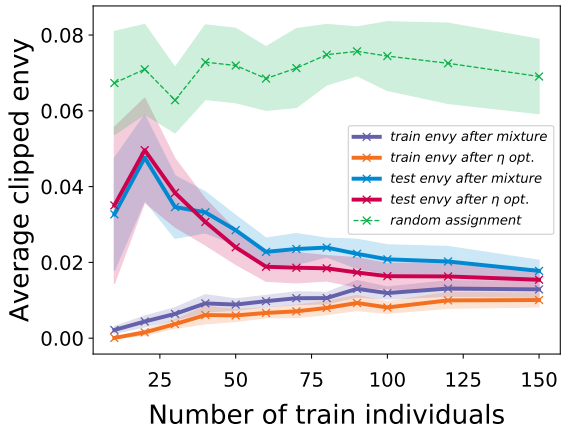


Figure 2.3: Training and test envy, as a function of the number of individuals. Shaded error bands depict 95% confidence intervals.

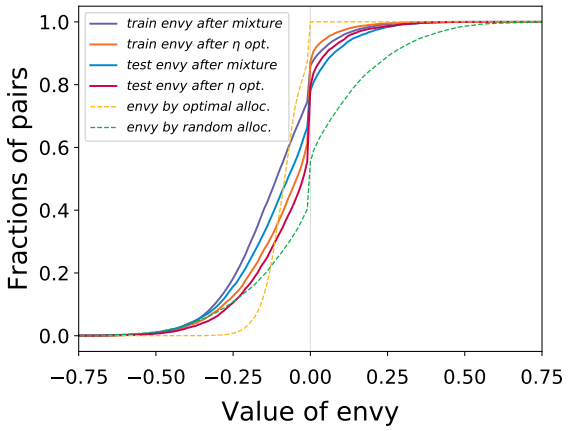


Figure 2.4: CDF of training and test envy for 100 training individuals

by a random allocation, which serves as a naïve benchmark for calibration purposes. Recall that the optimal assignment $h_{\vec{g}^\star,\eta^\star}$ has loss 0. For both the training and testing individuals, optimizing η improves the loss of the learned classifer. Moreover, our algorithms achieve low training errors for all dataset sizes, and as the dataset grows the testing error converges to the training error.

Figure 2.3 shows the average envy among pairs in the training data and test data, where, for each pair, negative envy is replaced with 0, to avoid obfuscating positive envy. The graph also depicts the average envy attained (on both the training and test data) by a random allocation. As for the losses, optimizing η results in lower average envy, and as the training set grows we see the generalization gap decrease.

In Figure 2.4 we zoom in on the case of 100 training individuals, and observe the empirical CDF of envy values. Interestingly, the optimal randomized classifier $h_{\vec{g}^\star,\eta^\star}$ shows lower negative envy values compared to other algorithms, but as expected has no positive envy pairs. Looking at the positive envy values, we can again see very encouraging results. In particular, for at least a 0.946 fraction of the pairs in the train data, we obtain envy of at most 0.05, and this generalizes to the test data, where for at least a 0.939 fraction of the pairs, we obtain envy of at most 0.1.

In summary, these results indicate that the algorithm described in Section 2.5.1 solves the optimization problem (2.1) for linear one-vs-all classifiers almost optimally, and that its output generalizes well even when the training set is small.

2.6 Conclusion

In this paper we propose EF as a suitable fairness notion for learning tasks with many outcomes over which individuals have heterogeneous preferences. We provide generalization

guarantees for a rich family of classifiers, showing that if we find a classifier that is envy-free on a sample of individuals, it will remain envy-free when we apply it to new individuals from the same distribution. This result circumvents an exponential lower bound on the sample complexity suffered by any two-stage learning algorithm that first finds an EF assignment for the sample and then extends it to the entire space. Finally, we empirically demonstrate that finding low-envy and low-loss classifiers is computationally tractable. These results show that envy-freeness is a practical notion of fairness for machine learning systems.

Part II

Voting and Social Choice

Chapter 3

Weighted Voting Via No-Regret Learning

Voting systems typically treat all voters equally. We argue that perhaps they should not: Voters who have supported good choices in the past should be given higher weight than voters who have supported bad ones. To develop a formal framework for desirable weighting schemes, we draw on *no-regret learning*. Specifically, given a voting rule, we wish to design a weighting scheme such that applying the voting rule, with voters weighted by the scheme, leads to choices that are almost as good as those endorsed by the best voter in hindsight. We derive possibility and impossibility results for the existence of such weighting schemes, depending on whether the voting rule and the weighting scheme are deterministic or randomized, as well as on the social choice axioms satisfied by the voting rule.

3.1 Introduction

In most elections, voters are entitled to equal voting power. This principle underlies the *one person, one vote* doctrine, and is enshrined in the United States Supreme Court ruling in the *Reynolds v. Sims* (1964) case.

But there are numerous voting systems in which voters do, in fact, have different *weights*. Standard examples include the European Council, where (for certain decisions) the weight of each member country is proportional to its population; and corporate voting procedures where stockholders have one vote per share. Some historical voting systems are even more pertinent: Sweden's 1866 system weighted voters by wealth, giving especially wealthy voters as many as 5000 votes; and a Belgian system, used for a decade at the end of the 19th Century, gave (at least) one vote to each man, (at least) two votes to each educated man, and three votes to men who were both educated and wealthy [Con11].

The last two examples can be seen as (silly, from a modern viewpoint) attempts to weight voters by *merit*, using wealth and education as measurable proxies thereof. We believe that the basic idea of weighting voters by merit does itself have merit. But we propose to measure a voter's merit by the *quality of his past votes*. That is, a voter who has supported good choices in the past should be given higher weight than a voter who has supported bad ones.

This high-level scheme is, arguably, most applicable to *repeated aggregation of objective opinions*. For example, consider a group of engineers trying to decide which prototype to develop, based on an objective measure of success such as projected market share. If an engineer supported a certain prototype and it turned out to be a success, she should be given higher weight compared to her peers in future decisions; if it is a failure, her weight should lower. Similar examples include a group of investors selecting companies to invest in; and a group of decision makers in a movie studio choosing movie scripts to produce. Importantly, the recently launched, not-for-profit website `RoboVote.org` already provides public access to voting tools for precisely these situations, albeit using methods that always treat all voters equally [PSZ16].

Our goal in this paper, therefore, is to augment existing voting methods with weights, in a way that keeps track of voters' past performance, and guarantees good choices over time. The main conceptual problem we face is the development of a formal framework in which one can reason about desirable weighting schemes.[1] To address this problem, we build on the *no-regret learning* literature, but depart from the classic setting in several ways — some superficial, and some fundamental.

Specifically, instead of experts, we have a set of n *voters*. In each round, each voter reveals a *ranking* over a set of alternatives, and the loss of each alternative is determined. In addition, we are given a (possibly randomized) *voting rule*, which receives weighted rankings as input, and outputs the winning alternative. The voting rule is not part of our design space; it is exogenous and fixed throughout the process. The *loss* of a voter in round t is given by assigning his ranking all the weight (equivalently, imagining that all voters have that ranking), applying the voting rule, and measuring the loss of the winning alternative (or the expected loss, if the rule is randomized). As in the classic setting, our benchmark is the best *voter* in hindsight (but we also discuss the stronger benchmark of best *voter weights* in hindsight in Section 3.6).

At first glance, it may seem that our setting easily reduces to the classic one, by treating voters as experts. But our loss is computed by applying the given voting rule to the entire profile of weighted rankings, and therein lies the rub.[2] This leads to our main research question:

> *For which voting rules is there a weighting scheme such that the difference between our average per-round loss and that of the best voter goes to zero as the number of rounds goes to infinity?*

In Section 3.4, we devise no-regret weighting schemes for any voting rule, under two classic feedback models — *full information* and *partial information*. While these results make no assumptions on the voting rule, the foregoing weighting schemes heavily rely on randomization. By contrast, deterministic weighting schemes seem more desirable, as they are easier to interpret and explain. In Section 3.5, therefore, we restrict our attention to deterministic weighting schemes. We find that if the voting rule is itself deterministic, it

[1] In that sense, our work is related to papers in *computational social choice* [Bra+16] that study weighted voting, in the context of manipulation, control, and bribery in elections [CSL07; ZPR09; FEL09; FEL15].

[2] For the same reason, our work is quite different from papers on online learning algorithms for ranking, where the algorithm chooses a ranking of objects at each stage, and suffers a loss based on the "relevance" of the ranking [RKJ08; CT15].

admits a no-regret weighting scheme if and only if it is *constant on unanimous profiles*. Because this property is not satisfied by any reasonable rule, the theorem should be interpreted as a strong impossibility result. We next consider randomized voting rules, and find that they give rise to much more subtle results, which depend on the properties of the voting rule in question.

3.2 Preliminaries

Our work draws on social choice theory and online learning. In this section we present important concepts and results from each of these areas in turn.

3.2.1 Social Choice

We consider a set $[n] \triangleq \{1, \dots, n\}$ of *voters* and a set A of m *alternatives*. A *vote* $\sigma : A \to [m]$ is a linear ordering — a ranking or permutation — of the alternatives. That is, for any vote σ and alternative a, $\sigma(a)$ denotes the position of alternative a in vote σ. For any $a, b \in A$, $\sigma(a) < \sigma(b)$ indicates that alternative a is preferred to b under vote σ. We also denote this preference by $a \succ_\sigma b$. We denote the set of all $m!$ possible votes over A by $\mathcal{L}(A)$.

A *vote profile* $\boldsymbol{\sigma} \in \mathcal{L}(A)^n$ denotes the votes of n voters. Furthermore, given a vote profile $\boldsymbol{\sigma} \in \mathcal{L}(A)^n$ and a weight vector $\mathbf{w} \in \mathbb{R}^n_{\geq 0}$, we define the *anonymous vote profile corresponding to* $\boldsymbol{\sigma}$ *and* $\mathbf{w}$, denoted $\boldsymbol{\pi} \in [0,1]^{|\mathcal{L}(A)|}$, by setting

$$\pi_\sigma \triangleq \frac{1}{\|\mathbf{w}\|_1} \sum_{i=1}^{n} w_i \mathbb{1}_{(\sigma_i = \sigma)}, \quad \forall \sigma \in \mathcal{L}(A).$$

That is, $\boldsymbol{\pi}$ is an $|\mathcal{L}(A)|$-dimensional vector such that for each vote $\sigma \in \mathcal{L}(A)$, π_σ is the fraction of the total weight on σ. When needed, we use $\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}$ to clarify the vote profile and weight vector to which the anonymous vote profile corresponds. Note that $\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}$ only contains the anonymized information about $\boldsymbol{\sigma}$ and $\mathbf{w}$, i.e., the anonymous vote profile remains the same even when the identities of the voters change.

To aggregate the (weighted) votes into a distribution over alternatives, we next introduce the concept of (anonymous) voting rules. Let $\Delta(\mathcal{L}(A))$ be the set of all possible anonymous vote profiles. Similarly, let $\Delta(A)$ denote the set of all possible distributions over A. An anonymous *voting rule* is a function $f : \Delta(\mathcal{L}(A)) \to \Delta(A)$ that takes as input an anonymous vote profile $\boldsymbol{\pi}$ and returns a distribution over the alternatives indicated by a vector $f(\boldsymbol{\pi})$, where $f(\boldsymbol{\pi})_a$ is the probability that alternative a is the winner under $\boldsymbol{\pi}$. We say that a voting rule f is *deterministic* if for any $\boldsymbol{\pi} \in \Delta(\mathcal{L}(A))$, $f(\boldsymbol{\pi})$ has support of size 1, i.e., there is a unique winner.

An anonymous voting rule f is called *strategyproof* if, informally, voters can never achieve a better outcome by misreporting their preferences (see Appendix B.1 for formal definitions). While strategyproofness is a natural property to be desired in a voting rule, the celebrated Gibbard-Satterthwaite Theorem [Gib73; Sat75] shows that non-dictatorial

strategyproof deterministic voting rules do not exist.[3] Subsequently, Gibbard [Gib77] extended this result to a characterization of strategyproof *randomized* voting rules. The next proposition is a direct corollary of his result for the case of anonymous rules.

Proposition 3.2.1. *Any strategyproof randomized rule is a distribution over a collection of the following types of rules:*

1. *Anonymous Unilaterals: g is an anonymous unilateral if there exists a function $h : \mathcal{L}(A) \to A$ for which*
$$g(\boldsymbol{\pi}) = \sum_{\sigma \in \mathcal{L}(A)} \pi_\sigma \mathbf{e}_{h(\sigma)},$$
where $\mathbf{e}_a$ is the unit vector that has 1 in the coordinate corresponding to $a \in A$, and 0 in all other coordinates.
2. *Duple: g is a duple rule if*
$$|\{a \in A \mid \exists \boldsymbol{\pi} \text{ such that } g(\boldsymbol{\pi})_a \neq 0\}| \leq 2.$$

Examples of strategyproof randomized voting rules include *randomized positional scoring rules* and the *randomized Copeland* rule, which were previously studied in this context [CS06; Pro10]. The reader is referred to Appendix B.1 for more details.

3.2.2 Online Learning

We next describe the general setting of online learning, also known as learning from experts. We consider a game between a *learner* and an *adversary*. There is a set of actions (a.k.a experts) $\mathcal{X}$ available to the learner, a set of actions $\mathcal{Y}$ available to the adversary, and a loss function $c : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ that is known to both parties. In every time step $t \in [T]$, the learner chooses a distribution, denoted by a vector $\mathbf{p}^t \in \Delta(\mathcal{X})$, over the actions in $\mathcal{X}$, and the adversary chooses an action y^t from the set $\mathcal{Y}$. The learner then receives a loss of $c(x^t, y^t)$ for $x^t \sim \mathbf{p}^t$. At this point, the learner receives some feedback regarding the action of the adversary. In the *full information* setting, the learner observes y^t before proceeding to the next time step. In the *partial information* setting, the learner only observes the loss $c(x^t, y^t)$.

The *regret* of the algorithm is defined as the difference between its total expected loss and that of the best fixed action in hindsight. The goal of the learner is to minimize its expected regret, that is, minimize

$$\mathbb{E}[Reg_T] \triangleq \mathbb{E}\left[\sum_{t=1}^{T} c(x^t, y^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} c(x, y^t)\right],$$

where the expectation is taken over the choice of $x^t \sim \mathbf{p}^t$, and any other random choices made by the algorithm and the adversary. An online algorithm is called a *no-regret* algorithm if $\mathbb{E}[Reg_T] \in o(T)$. In words, the average regret of the learner must go to 0 as $T \to \infty$. In general, deterministic algorithms, for which $\|\mathbf{p}^t\|_\infty = 1$, can suffer linear regret, because

[3] The theorem also requires a range of size at least 3.

the adversary can choose a sequence of actions $y^1, \ldots, y^T$ on which the algorithm makes sub-optimal decisions at every round. Therefore, randomization is one of the key aspects of no-regret algorithms.

Many online no-regret algorithms are known for the full information and the partial information settings. In particular, the HEDGE algorithm [FS95] is one of the earliest results in this space for the full information setting. At time $t+1$, HEDGE picks each action x with probability $p_x^{t+1} \propto \exp(-\eta C^t(x))$, for $C^t(x) = \sum_{s=1}^{t} c(x, y^s)$ and $\eta = \Theta(\sqrt{2\ln(|\mathcal{X}|)\,/\,T})$.

Proposition 3.2.2 (Freund and Schapire 1995)**.** HEDGE *has regret*

$$\mathbb{E}[Reg_T] \leq O\left(\sqrt{T \ln(|\mathcal{X}|)}\right).$$

For the partial information setting, the EXP3 algorithm of Auer, Cesa-Bianchi, Freund, and Schapire [Aue+02] can be thought of as a variant of the HEDGE algorithm with importance weighting. In particular, at time $t+1$, EXP3 picks each action x with probability $p_x^{t+1} \propto \exp(-\eta \tilde{C}^t(x))$, for $\eta = \Theta(\sqrt{2\ln(|\mathcal{X}|)\,/\,T|\mathcal{X}|})$ and

$$\tilde{C}^t(x) = \sum_{s=1}^{t} \frac{\mathbb{1}_{(x^s=x)} c(x, y^s)}{p_x^s}. \tag{3.1}$$

In other words, EXP3 is similar to HEDGE, except that instead of taking into account the total loss of an action, $C^t(x)$, it takes into account an *estimate* of the loss, $\tilde{C}^t(x)$.

3.3 Problem Formulation

In this section, we formulate the question of how one can design a weighting scheme that effectively weights the rankings of voters based on the history of their votes and the performance of the selected alternatives.

We consider a setting where n voters participate in a sequence of elections that are decided by a known voting rule f. In each election, voters submit their rankings over a different set of m alternatives so as to elect a winner. Given an adversarial sequence of voters' rankings $\boldsymbol{\sigma}^{1:T}$ and alternative losses $\boldsymbol{\ell}^{1:T}$ over a span of T elections, the best voter is the one whose rankings lead to the election of the winners with smallest loss overall. We call this voter *the best voter in hindsight*. (See Section 3.6 for a discussion of a stronger benchmark: best *weight vector* in hindsight.)

When the sequence of elections is not known a priori, the best voter is not known either. In this case, the weighting scheme has to take an online approach to weighting the voters' rankings. That is, at each time step $t \leq T$, the weighting scheme chooses a weight vector $\mathbf{w}^t$, possibly at random, to weight the rankings of the voters. After the election is held, the weighting scheme receives some feedback regarding the quality of the alternatives in that election, typically in the form of the loss of the elected alternative or that of all alternatives. Using the feedback, the weighting scheme then re-weights the voters' rankings based on their performance so far. Our goal is to design a weighting scheme that weights the rankings of the voters at each time step, and elects winners with overall expected loss

that is almost as small as that of the best voter in hindsight. We refer to the expected difference between these losses as the expected *regret*. Formally, let

$$L_f(\boldsymbol{\pi}, \boldsymbol{\ell}) \triangleq \sum_{a \in A} f(\boldsymbol{\pi})_a \cdot \ell_a$$

be the expected loss of the (possibly randomized) voting rule f under the anonymous preference profile $\boldsymbol{\pi}$ and loss vector $\boldsymbol{\ell}$. Then the expected regret is

$$\mathbb{E}[Reg_T] \triangleq \mathbb{E}\left[\sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}, \boldsymbol{\ell}^t) - \min_i \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t)\right],$$

where the expectation is taken over any additional source of randomness in the adversarial sequence or the algorithm. In particular, we seek a weighting scheme for which the average expected regret goes to zero as the time horizon T goes to infinity, at a rate that is polynomial in the number of voters and alternatives. That is, we wish to achieve $\mathbb{E}[Reg_T] = \text{poly}(n, m) \cdot o(T)$. This is our version of a *no-regret* algorithm.

The type of the feedback is an important factor in designing a weighting scheme. Analogously to the online learning models described in Section 3.2.2, we consider two types of feedback, *full information* and *partial information*. In the full information case, after a winner is selected at time t, the quality of all alternatives and rankings of the voters at that round are revealed to the weighting scheme. Note that this information is sufficient for computing the loss of each voter's rankings so far. This would be the case, for example, if the alternatives are companies to invest in. On the other hand, in the partial information setting only the loss of the winner is revealed. This type of feedback is appropriate when the alternatives are product prototypes: we cannot know how successful an undeveloped prototype would have been, but obviously we can measure the success of a prototype that was selected for development. More formally, in the full information setting the choice of $\mathbf{w}^{t+1}$ can depend on $\boldsymbol{\sigma}^{1:t}$ and $\boldsymbol{\ell}^{1:t}$, while in the partial information setting it can only depend on $\boldsymbol{\sigma}^{1:t}$ and $\ell^s_{a^s}$ for $s \leq t$, where a^s is the alternative that won the election at time s.

No doubt the reader has noted that the above problem formulation is closely related to the general setting of online learning. Using the language of online learning introduced in Section 3.2.2, the weight vector $\mathbf{w}^t$ corresponds to the learner's action x^t, the vote profile and alternative losses $(\boldsymbol{\sigma}^t, \boldsymbol{\ell}^t)$ correspond to the adversary's action y^t, the expected loss of the weighting scheme $L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}, \boldsymbol{\ell}^t)$ corresponds to the loss of the learning algorithm $c(x^t, y^t)$, and the best-in-hindsight voter — or weight vector $\mathbf{e}_i$ — refers to the best-in-hindsight action.

3.4 Randomized Weights

In this section, we develop no-regret algorithms for the full information and partial information settings. We essentially require no assumptions on the voting rule, but also impose no restrictions on the weighting scheme. In particular, the weighting scheme may be randomized, that is, the weights can be sampled from a distribution over weight vectors. This allows us to obtain general positive results.

As we just discussed, our setting is closely related to the classic online learning setting. Here, we introduce an algorithm analogous to HEDGE that works in the full information setting of Section 3.3 and achieves no-regret guarantees.

Algorithm 3.1: Full information setting, using randomized weights.

Input: Adversarial sequences $\boldsymbol{\sigma}^{1:T}$ and $\boldsymbol{\ell}^{1:T}$, and parameter $\eta = \sqrt{2\ln n/T}$

for $t = 1, \ldots, T$ **do**

  Play weight vector $\mathbf{e}_i$ with probability

$$p_i^t \propto \exp\left(-\eta \sum_{s=1}^{t-1} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^s, \mathbf{e}_i}, \boldsymbol{\ell}^s)\right).$$

  Observe $\boldsymbol{\ell}^t$ and $\boldsymbol{\sigma}^t$.

end

Theorem 3.4.1. *For any anonymous voting rule f and n voters, Algorithm 3.1 has regret $O(\sqrt{T\ln(n)})$ in the full information setting.*

Proof sketch. At a high level, this algorithm only considers weight vectors that correspond to a single voter. At every time step, the algorithm chooses a distribution over such weight vectors and applies the voting rule to one such weight vector that is drawn at random from this distribution. This is equivalent to applying the HEDGE algorithm to a set of actions, each of which is a weight vector that corresponds to a single voter. In addition, the loss of the benchmark weighting scheme is the smallest loss that one can get from following one such weight vector. Therefore, the theorem follows from Proposition 3.2.2. □

Let us now address the partial information setting. One may wonder whether the above approach, i.e., reducing our problem to online learning and using a standard algorithm, directly extends to the partial information setting (with the EXP3 algorithm). The answer is that it does not. In particular, in the classic setting of online learning with partial information feedback, the algorithm can compute the estimated loss of the action it just played, that is, the algorithm can compute $c(x^t, y^t)$. In our problem setting, however, the weighting scheme only observes $\boldsymbol{\sigma}^t$ and $\ell^t_{a^t}$ for the specific alternative a^t that was elected at this time. Since the losses of other alternatives remain unknown, the weighting scheme cannot even compute the expected loss of the specific voter i^t it selected at time t, i.e., $L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}}, \boldsymbol{\ell}^t)$. Therefore, we cannot directly use the EXP3 algorithm by imagining that the voters are actions, as we do not obtain the partial information feedback that the algorithm requires. Nevertheless, we can design a new algorithm inspired by EXP3.

Theorem 3.4.2. *For any anonymous voting rule f and n voters, Algorithm 3.2 has regret $O(\sqrt{Tn\ln(n)})$ in the partial information setting.*

To prove the theorem, we show that certain properties, which are necessary for the performance of EXP3, still hold in our setting. Specifically, Lemma 3.4.3 asserts that $\tilde{\boldsymbol{\ell}}^t$ creates an unbiased estimator of the expected loss of the weighting scheme. Moreover, it

Algorithm 3.2: Partial information setting, using randomized weights.

Input: An adversarial sequences of $\boldsymbol{\sigma}^{1:T}$ and $\boldsymbol{\ell}^{1:T}$, and parameter $\eta = \sqrt{2 \ln n / Tn}$.

Let $\tilde{\mathbf{L}}^0 = \mathbf{0}$.
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, n$ **do**
    Let $p_i^t \propto \exp(-\eta \tilde{L}_i^{t-1})$.
  end
  Play weight vector $\mathbf{e}_{i^t}$ from distribution $\mathbf{p}^t$, and observe the vote profile $\boldsymbol{\sigma}^t$, the alternative $a^t \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}})$, and its loss $\ell_{a^t}^t$.
  Let $\tilde{\boldsymbol{\ell}}^t$ be the vector such that $\tilde{\ell}_{i^t}^t = \ell_{a^t}^t / p_{i^t}^t$ and $\tilde{\ell}_i^t = 0$ for $i \neq i^t$.
  Let $\tilde{\mathbf{L}}^t = \tilde{\mathbf{L}}^{t-1} + \tilde{\boldsymbol{\ell}}^t$.
end

states that for any voter i^*, $\tilde{L}_{i^*}^t$ is an unbiased estimator for the loss that the weighting scheme would have received if it followed the rankings of voter i^* throughout the sequence of elections. Lemma 3.4.4 then establishes that the variance of this estimator is small.

Lemma 3.4.3. *For any t, any i^*, $i^t \sim \mathbf{p}^t$, and $a^t \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}})$, we have*

$$\mathbb{E}_{i^t, a^t}\left[\sum_{i=1}^{n} p_i^t \tilde{\ell}_i^t\right] = \mathbb{E}_{i^t}\left[L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}}, \boldsymbol{\ell}^t)\right]$$

and

$$\mathbb{E}_{i^t, a^t}\left[\tilde{L}_{i^*}^T\right] = \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^*}}, \boldsymbol{\ell}^t).$$

Proof. For ease of notation, we suppress t when it is clear from the context. First note that $\tilde{\boldsymbol{\ell}}$ is zero in all of its elements, except for $\tilde{\ell}_{i^t}$. So,

$$\sum_{i=1}^{n} p_i \tilde{\ell}_i = p_{i^t} \tilde{\ell}_{i^t} = p_{i^t} \frac{\ell_{a^t}}{p_{i^t}} = \ell_{a^t}.$$

Therefore, we have

$$\mathbb{E}_{i^t, a^t}\left[\sum_{i=1}^{n} p_i \tilde{\ell}_i\right] = \mathbb{E}_{i^t, a^t}\left[\ell_{a^t}\right] = \mathbb{E}_{i^t}\left[L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}, \mathbf{e}_{i^t}}, \boldsymbol{\ell})\right].$$

For clarity of presentation, let $\tilde{\boldsymbol{\ell}}^{i,a}$ be an alternative representation of $\tilde{\boldsymbol{\ell}}$ when $i^t = i$ and $a^t = a$. Note that $\ell_{i^*}^{i,a} \neq 0$ only if $i^* = i$. We have

$$\mathbb{E}_{i^t, a^t}\left[\tilde{L}_{i^*}^T\right] = \sum_{t=1}^{T} \mathbb{E}_{i^t, a^t}\left[\tilde{\ell}_{i^*}^{i^t, a^t}\right]$$

$$
\begin{aligned}
&= \sum_{t=1}^{T} \sum_{i=1}^{n} p_i^t \, \mathbb{E}_{a \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i})} \left[\tilde{\ell}_{i^*}^{i,a} \right] \\
&= \sum_{t=1}^{T} p_{i^*}^t \, \mathbb{E}_{a \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^*}})} \left[\frac{\ell_a^t}{p_{i^*}^t} \right] \\
&= \sum_{t=1}^{T} \mathbb{E}_{a \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^*}})} \left[\ell_a^t \right] \\
&= \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^*}}, \boldsymbol{\ell}^t).
\end{aligned}
$$

□

Lemma 3.4.4. *For any t, $i^t \sim \mathbf{p}^t$, and $a^t \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}})$, we have*

$$
\mathbb{E}_{i^t, a^t} \left[\sum_{i=1}^{n} p_i^t (\tilde{\ell}_i^t)^2 \right] \leq n.
$$

Proof. For ease of notation, we suppress t when it is clear from the context. Since $\tilde{\boldsymbol{\ell}}$ is zero in all of its elements, except for $\tilde{\ell}_{i^t}$, we have

$$
\sum_{i=1}^{n} p_i (\tilde{\ell}_i)^2 = p_{i^t} (\tilde{\ell}_{i^t})^2 = p_{i^t} \left(\frac{\ell_{a^t}}{p_{i^t}} \right)^2 = \frac{(\ell_{a^t})^2}{p_{i^t}}.
$$

Therefore,

$$
\begin{aligned}
\mathbb{E}_{i^t, a^t} \left[\sum_{i=1}^{n} p_i (\tilde{\ell}_i)^2 \right] &= \mathbb{E}_{i^t, a^t} \left[\frac{(\ell_{a^t})^2}{p_{i^t}} \right] \\
&= \sum_{i=1}^{n} p_i \, \mathbb{E}_{a \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}, \mathbf{e}_i})} \left[\frac{(\ell_a)^2}{p_i} \right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{a \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}, \mathbf{e}_i})} \left[(\ell_a)^2 \right] \\
&\leq n.
\end{aligned}
$$

□

We are now ready to prove the theorem.

Proof of Theorem 3.4.2. We use a potential function, given by $\Phi^t \triangleq -\frac{1}{\eta} \ln \left(\sum_{i=1}^{n} \exp(-\eta \tilde{L}_i^{t-1}) \right)$. We prove the claim by analyzing the expected increase in this potential function at every

time step. Note that

$$\begin{aligned}\Phi_{t+1} - \Phi_t &= -\frac{1}{\eta} \ln \left(\frac{\sum_{i=1}^n \exp(-\eta \tilde{L}_i^{t-1} - \eta \tilde{\ell}_i^t)}{\sum_{i=1}^n \exp(-\eta \tilde{L}_i^{t-1})} \right) \\ &= -\frac{1}{\eta} \ln \left(\sum_{i=1}^n p_i^t \exp(-\eta \tilde{\ell}_i^t) \right).\end{aligned} \tag{3.2}$$

Taking the expected increase in the potential function over the random choices of i^t and a^t for all $t = 1, \ldots, T$, we have

$$\begin{aligned}&\mathbb{E}\left[\Phi_{T+1} - \Phi_1\right] \\ &= \sum_{t=1}^T \mathbb{E}_{i^t, a^t}\left[\Phi_{t+1} - \Phi_t\right] \\ &\geq \sum_{t=1}^T \mathbb{E}_{i^t, a^t}\left[-\frac{1}{\eta} \ln \left(\sum_{i=1}^n p_i^t \left(1 - \eta \tilde{\ell}_i^t + \frac{1}{2} \left(\eta \tilde{\ell}_i^t \right)^2 \right) \right) \right] \\ &= \sum_{t=1}^T \mathbb{E}_{i^t, a^t}\left[-\frac{1}{\eta} \ln \left(1 - \eta \left(\sum_{i=1}^n p_i^t \tilde{\ell}_i^t - \frac{\eta}{2} \sum_{i=1}^n p_i^t \left(\tilde{\ell}_i^t \right)^2 \right) \right) \right] \\ &\geq \sum_{t=1}^T \mathbb{E}_{i^t, a^t}\left[\sum_{i=1}^n p_i^t \tilde{\ell}_i^t - \frac{\eta}{2} \sum_{i=1}^n p_i^t \left(\tilde{\ell}_i^t \right)^2 \right] \\ &\geq \mathbb{E}\left[\sum_{t=1}^T L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}}, \boldsymbol{\ell}^t) \right] - \frac{\eta T n}{2},\end{aligned} \tag{3.3}$$

where the second transition follows from Equation (3.2) because for all $x \geq 0$, $e^{-x} \leq 1 - x + \frac{x^2}{2}$, the fourth transition follows from $\ln(1-x) \leq -x$ for all $x \in \mathbb{R}$, and the last transition holds by Lemmas 3.4.3 and 3.4.4. On the other hand, $\Phi_1 = -\frac{1}{\eta} \ln n$ and for any i^*,

$$\Phi_{T+1} \leq -\frac{1}{\eta} \ln \left(\exp(-\eta \tilde{L}_{i^*}^T) \right) = \tilde{L}_{i^*}^T.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\Phi_{T+1} - \Phi_1\right] &\leq \mathbb{E}\left[\tilde{L}_{i^*}^T + \frac{1}{\eta} \ln n \right] \\ &= \mathbb{E}\left[\sum_{t=1}^T L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^*}}, \boldsymbol{\ell}^t) + \frac{1}{\eta} \ln n \right].\end{aligned} \tag{3.4}$$

We can now prove the theorem by using Equations (3.3) and (3.4), and the parameter value $\eta = \sqrt{2 \ln n / Tn}$:

$$\mathbb{E}\left[\sum_{t=1}^T L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_{i^t}}, \boldsymbol{\ell}^t) - \min_{i \in [n]} \sum_{t=1}^T L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t) \right]$$

$$\leq \frac{1}{\eta}\ln n + \frac{\eta T n}{2}$$
$$\leq \sqrt{2Tn\ln n}.$$

□

3.5 Deterministic Weights

One of the key aspects of the weighting schemes we used in the previous section is randomization. In such weighting schemes, the weights of the voters not only depend on their performance so far, but also on the algorithm's coin flips. In practice, voters would most likely prefer weighting schemes that depend only on their past performance, and are therefore easier to interpret.

In this section, we focus on designing weighting schemes that are deterministic in nature. Formally, a *deterministic weighting scheme* is an algorithm that at time step $t+1$ deterministically chooses one weight vector $\mathbf{w}^{t+1}$ based on the history of play, i.e., sequences $\boldsymbol{\sigma}^{1:t}$, $\boldsymbol{\ell}^{1:t}$, and $a^{1:t}$. In this section, we seek an answer to the following question: *For which voting rules is there a no-regret deterministic weighting scheme?* In contrast to the results established in the previous section, we find that the properties of the voting rule play an important role here. In the remainder of this section, we show possibility and impossibility results for the existence of such weighting schemes under randomized and deterministic voting rules.

We begin our search for deterministic weighting schemes by considering deterministic voting rules. Note that in this case the winning alternatives are induced deterministically by the weighting scheme, so the weight vector $\mathbf{w}^{t+1}$ should be deterministically chosen based on the sequences $\boldsymbol{\sigma}^{1:t}$ and $\boldsymbol{\ell}^{1:t}$. We establish an impossibility result: Essentially no deterministic weighting scheme is no-regret for a deterministic voting rule. Specifically, we show that a deterministic no-regret weighting scheme exists for a deterministic voting rule if and only if the voting rule is constant on unanimous profiles.

Definition 3.5.1. A voting rule f is *constant on unanimous profiles* if and only if for all $\sigma, \sigma' \in \mathcal{L}(A)$, $f(\mathbf{e}_\sigma) = f(\mathbf{e}_{\sigma'})$, where $\mathbf{e}_\sigma$ denotes the anonymous vote profile that has all of its weight on ranking σ.

Theorem 3.5.2. *For any deterministic voting rule f, a deterministic weighting scheme with regret $o(T)$ exists if and only if f is constant on unanimous profiles. This is true in both the full information and partial information settings.*

Proof. We first prove that for any voting rule that is constant on unanimous profiles there exists a deterministic weighting scheme that is no-regret. Consider such a voting rule f and a simple deterministic weighting scheme that uses weight vector $\mathbf{w}^t = \mathbf{e}_1$ for every time step $t \leq T$ (so it does not use feedback — whether full or partial — at all). Note that at each time step t and for any voter $i \in [n]$,

$$f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}) = f(\mathbf{e}_{\sigma_1^t}) = f(\mathbf{e}_{\sigma_i^t}) = f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}),$$

where the second transition holds because f is constant on unanimous profiles. As a result,

$$L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}, \boldsymbol{\ell}^t) = L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}, \boldsymbol{\ell}^t).$$

In words, the total loss of the weighting scheme is the same as the total loss of any individual voter — this weighting scheme has 0 regret.

Next, we prove that if f is not constant on unanimous profiles then for any deterministic weighting scheme there is an adversarial sequence of $\boldsymbol{\sigma}^{1:T}$ and $\boldsymbol{\ell}^{1:T}$ that leads to regret of $\Omega(T)$, even in the full information setting. Take any such voting rule f and let $\tau, \tau' \in \mathcal{L}(A)$ be such that $f(\mathbf{e}_\tau) \neq f(\mathbf{e}_{\tau'})$. At time t, the adversary chooses $\boldsymbol{\sigma}^t$ and $\boldsymbol{\ell}^t$ based on the deterministic weight vector $\mathbf{w}^t$ as follows: The adversary sets $\boldsymbol{\sigma}^t$ to be such that $\sigma^t_1 = \tau$ and $\sigma^t_j = \tau'$ for all $j \neq 1$. Let alternative a^t be the winner of profile $\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}$, i.e., $f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}) = \mathbf{e}_{a^t}$. The adversary sets $\ell^t_{a^t} = 1$ and $\ell^t_x = 0$ for all $x \neq a^t$. Therefore, the weighting scheme incurs a loss of 1 at every step, and its total loss is

$$\sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}, \boldsymbol{\ell}^t) = \sum_{t=1}^{T} \ell^t_{a^t} = T.$$

Let us consider the total loss that the ranking of any individual voter incurs. By design, for any $j > 1$,

$$f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_1}) = f(\mathbf{e}_\tau) \neq f(\mathbf{e}_{\tau'}) = f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_j}).$$

Therefore, for at least one voter $i \in [n]$, $f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}) \neq \mathbf{e}_{a^t}$. Note that such a voter receives loss of 0, so the combined loss of all voters is at most $n-1$. Over all time steps, the total combined loss of all voters is at most $T(n-1)$. As a result, the best voter incurs a loss of at most $\frac{(n-1)T}{n}$, i.e., the average loss. We conclude that the regret of the weighting scheme is

$$\begin{aligned} Reg_T &= \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}, \boldsymbol{\ell}^t) - \min_{i\in[n]} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}, \boldsymbol{\ell}^t) \\ &\geq T - \frac{(n-1)T}{n} \\ &= \frac{T}{n}. \end{aligned}$$

□

Theorem 3.5.2 indicates that we need to allow randomness (either in the weighting scheme or in the voting rule) if we wish to have no-regret guarantees. As stated before, we would like to have a deterministic weighting scheme so that the weights of voters are not decided by coin flips. This leaves us with no choice other than having a randomized voting rule. Nonetheless, one might argue in favor of having a deterministic voting rule and a randomized weighting scheme, claiming that it is equivalent because the randomness has simply been shifted from the voting rule to the weights. To that imaginary critic we say that allowing the voting rule to be randomized makes it possible to achieve strategyproofness (see Section 3.2.1), which cannot be satisfied by a deterministic voting rule.

We next show that for any voting rule that is a distribution over unilaterals there exist deterministic weighting schemes that are no-regret. An important family of strategyproof randomized voting rules — randomized positional scoring rules (see Appendix B.1) — can be represented as distributions over unilaterals, hence the theorem allows us to design a no-regret weighting scheme for any randomized positional scoring rule.

The weighting schemes that we use build on Algorithms 3.1 and 3.2 directly. In more detail, we consider deterministic weighting schemes that at time t use weight vector $\mathbf{p}^t$ and a randomly drawn candidate $a^t \sim f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{p}^t})$, where $\mathbf{p}^t$ is computed according to Algorithms 3.1 or 3.2. The key insight behind these weighting schemes is that, if f is a distribution over unilaterals, we have

$$\mathbb{E}_{i\sim\mathbf{p}^t}[f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i})] = f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{p}^t}), \tag{3.5}$$

where the left-hand side is a vector of expectations. That is, the outcome of the voting rule $f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{p}^t})$ can be alternatively implemented by applying the voting rule on the ranking of voter i that is drawn at random from the distribution $\mathbf{p}^t$. This is exactly what Algorithms 3.1 and 3.2 do. Therefore, the deterministic weighting schemes induce the same distribution over alternatives at every time step as their randomized counterparts, and achieve the same regret. The next theorem, whose full proof appears in Appendix B.2, formalizes this discussion.

Theorem 3.5.3. *For any voting rule that is a distribution over unilaterals, there exist deterministic weighting schemes with regret of $O(\sqrt{T\ln(n)})$ and $O(\sqrt{Tn\ln(n)})$ in the full-information and partial-information settings, respectively.*

The theorem states that there exist no-regret deterministic weighting schemes for any voting rule that is a distribution over unilaterals. It is natural to ask whether being a distribution over unilaterals is, in some sense, also a necessary condition. While we do not give a complete answer to this question, we are able to identify a sufficient condition for *not* having no-regret deterministic weighting schemes.

To this end, we introduce a classic concept. Alternative $a \in A$ is a *Condorcet winner* in a given vote profile if for every $b \in A$, a majority of voters rank a above b. A deterministic rule is *Condorcet consistent* if it selects a Condorcet winner whenever one exists in the given vote profile; see Appendix B.1 for formal definitions. We extend the notion of Condorcet consistency to randomized rules.

Definition 3.5.4. For a set of alternatives A such that $|A| = m$, a randomized voting rule $f : \Delta(\mathcal{L}(A)) \to \Delta(A)$ is *probabilistically Condorcet consistent with gap $\delta(m)$* if for any anonymous vote profile $\boldsymbol{\pi}$ that has a Condorcet winner a, and for all alternatives $x \in A \setminus \{a\}$, $f(\boldsymbol{\pi})_a \geq f(\boldsymbol{\pi})_x + \delta(m)$.

In words, a randomized voting rule is probabilistically Condorcet consistent if the Condorcet winner has strictly higher probability of being selected than any other alternative, by a gap of $\delta(m)$. As an example, a significant strategyproof randomized voting rule — the randomized Copeland rule, defined in Appendix B.1 — is probabilistically Condorcet consistent with gap $\delta(m) = \Omega(1/m^2)$.

Theorem 3.5.5. *For a set of alternatives A such that $|A| = m$, let f be a probabilistically*

Condorcet consistent voting rule with gap $\delta(m)$, and suppose there are n voters for

$$n \geq 2\left(\frac{3}{2\delta(m)} + 1\right).$$

Then any deterministic weighting scheme will suffer regret of $\Omega(T)$ under f (in the worst case), even in the full information setting.

The theorem's proof is relegated to Appendix B.3. It is interesting to note that Theorems 3.5.3 and 3.5.5 together imply that distributions over unilaterals are not probabilistically Condorcet consistent. This is actually quite intuitive: Distributions over unilaterals are "local" in that they look at each voter separately, whereas Condorcet consistency is a global property. In fact, these theorems can be used to prove — in an especially convoluted and indirect way — a simple result from social choice theory [Mou83]: No positional scoring rule is Condorcet consistent!

3.6 Discussion

We conclude by discussing several conceptual points.

A natural, stronger benchmark. In our model (see Section 3.3), we are competing with the best voter in hindsight. But our action space consists of *weight vectors*. It is therefore natural to ask whether we can compete with the best weight vector in hindsight (hereinafter, the *stronger benchmark*). Clearly the stronger benchmark is indeed at least as hard, because the best voter i^* corresponds to the weight vector $\mathbf{e}_{i^*}$. Therefore, our impossibility results for competing against the best voter in hindsight (Theorems 3.5.2 and 3.5.5) extend to the stronger benchmark. Moreover, voting rules that are distributions over unilaterals demonstrate a certain linear structure where the outcome of the voting rule nicely decomposes across individual voters. Under such voting rules, the benchmark of best weights in hindsight is equivalent to the benchmark of best voter in hindsight. Therefore, Theorem 3.5.3 also holds for the stronger benchmark, and, in summary, each and every result of Section 3.5 extends to the stronger benchmark. By contrast, Theorems 3.4.1 and 3.4.2 do not hold for the stronger benchmark; the question of identifying properties of voting rules (beyond distributions over unilaterals) that admit *randomized* no-regret weighting schemes under the stronger benchmark remains open. We describe the stronger benchmark in more detail, and formalize the above arguments, in Appendix B.4.

Changing the sets of alternatives and voters over time. We wish to emphasize that the set of alternatives at each time step, i.e., in each election, can be completely different. Moreover, the *number* of alternatives could be different. In fact, our positive results do not even depend on the number of alternatives m, so we can simply set m to be an upper bound. By contrast, we do need the set of voters to stay fixed throughout the process, but this is consistent with our motivating examples (e.g., a group of partners in a small venture capital firm would face different choices at every time step, but the composition of the group rarely changes).

Optimizing the voting rule. Throughout the paper, the voting rule is exogenous. One might ask whether it makes sense to optimize the choice of voting rule itself, in order to obtain good no-regret learning results. Our answer is "yes and no". On the one hand, we believe our results do give some guidance on choosing between voting rules. For example, from this viewpoint, one might prefer randomized Borda (which admits no-regret algorithms under a deterministic weighting scheme) to randomized Copeland (which does not). On the other hand, many considerations are factored into the choice of voting rule: social choice axioms, optimization of additional objectives [PSZ16; Bou+15; EFS09; CS05], and simplicity. It is therefore best to think of our approach as *augmenting* voting rules that are already in place.

Chapter 4

Virtual Democracy: A Voting-Based Framework for Automating Decisions

We present a general approach to automating decisions, drawing on machine learning and computational social choice. In a nutshell, we propose to *learn* a model of societal preferences, and, when faced with a specific dilemma at runtime, efficiently *aggregate* those preferences to identify a desirable choice. We provide a concrete algorithm that instantiates our approach; some of its crucial steps are informed by a new theory of *swap-dominance efficient* voting rules. Finally, as a proof of concept, we implement and evaluate a system for decision making in the autonomous vehicle domain, using preference data collected from 1.3 million people through the Moral Machine website.

4.1 Introduction

One of the most basic ideas underlying democracy is that complicated decisions can be made by asking a group of people to vote on the alternatives at hand. As a decision-making framework, this paradigm is versatile, because people can express a sensible opinion about a wide range of issues. One of its seemingly inherent shortcomings, though, is that voters must take the time to cast a vote — hopefully an informed one — every time a new dilemma arises.

Consider the following example. A group of friends have to decide where to have lunch every day. They achieve this by voting on the set of restaurants available each day, and then aggregating these votes to pick the winning restaurant. But, this can be tedious if it has to be repeated every single day, as each voter would have to reconsider all the available choices that day, take into account where they ate the previous day, what cuisines they have eaten in the previous week, how far or expensive each option is, and so on, and then construct a complete ranking over all the available options. What if we could instead *predict* the preferences of the voters — instead of explicitly asking them to vote — and then aggregate those predicted preferences to arrive at a decision? In particular, we could learn a virtual model for each of the voters, and let these models vote on the voters' behalf.

This would automate the system and help us make decisions more efficiently.

Even though the example illustrates the automation of a voting scenario, this approach can be used much more generally to automate decision making whenever aggregating people's opinions is justifiable. Further, this is especially useful when there is a lack of a formal specification of ground-truth principles to be followed in order to make the decision. In such cases, the decision can be automated by aggregating people's opinions on the specific delimmas at hand. And the key idea behind this approach is that instead of consulting each of the voters on each decision (which would have been the ideal scenario), we could automate these decisions by learning a model for each of the voters, and then using these models as a proxy for the flesh and blood voters. In other words, the models serve as virtual voters, which is why we refer to this paradigm as *virtual democracy*.

As a proof of concept, we also apply this approach to an enormous dataset collected through the website Moral Machine.[1] This website presents a modern variant of the classic *trolley problem* [Jar85]: An autonomous vehicle has a brake failure, leading to an accident with inevitably tragic consequences; due to the vehicle's superior perception and computation capabilities, it can make an informed decision. Should it stay its course and hit a wall, killing its three passengers, one of whom is a young girl? Or swerve and kill a male athlete and his dog, who are crossing the street on a red light? A notable paper by Bonnefon, Shariff, and Rahwan [BSR16] has shed some light on how people address such questions, and even former US President Barack Obama has weighed in.[2]

More concretely, our approach for automating decision making consists of four steps, drawing on machine learning and *computational social choice* [Bra+16]:[3]

I *Data collection:* Ask human voters to compare pairs of alternatives (say a few dozen per voter). In the autonomous vehicle domain, an alternative is determined by a vector of features such as the number of victims and their gender, age, health — even species!

II *Learning:* Use the pairwise comparisons to learn a model of the preferences of each voter over all possible alternatives.

III *Summarization:* Combine the individual models into a single model, which approximately captures the collective preferences of all voters over all possible alternatives.

IV *Aggregation:* At runtime, when encountering a dilemma involving a specific subset of alternatives, use the summary model to deduce the preferences of all voters over this particular subset, and apply a voting rule to aggregate these preferences into a collective decision. In the autonomous vehicle domain, the selected alternative is the outcome that society (as represented by the voters whose preferences were elicited in Step I) views as the least catastrophic among the grim options the vehicle currently faces.

Note that we are not advocating the use of this approach as is for the autonomous vehicle domain in real life, but only using the corresponding dataset as a proof of concept

[1] `http://moralmachine.mit.edu`

[2] `https://www.wired.com/2016/10/president-obama-mit-joi-ito-interview/`

[3] which deals with algorithms for aggregating individual preferences towards collective decisions

for the approach. More on this in Section 4.6.

For Step I, the pairwise comparisons would generally be collected from voters who would be affected or involved in the whole decision making process at hand. For the autonomous vehicle domain, we note that it is possible to collect an adequate dataset through, say, Amazon Mechanical Turk. But we actually perform this step on a much larger scale. Indeed, we use, for the first time, a unique dataset that consists of 18,254,285 pairwise comparisons between alternatives in the autonomous vehicle domain, obtained from 1,303,778 voters, through the website Moral Machine.

Subsequent steps (namely Steps II, III, and IV) rely on having a *model* for preferences. There is a considerable line of work on distributions over rankings over a *finite* set of alternatives. A popular class of such models is the class of *random utility models* [APX12; APX14b; MG15; GS09; RA14], which use random utilities for alternatives to generate rankings over the alternatives. We require a slightly more general notion, as we are interested in situations where the set of alternatives is infinite, and any finite subset of alternatives might be encountered (c.f. Caron and Teh 2012). For example, there are uncountably many scenarios an autonomous vehicle might face, because one can choose to model some features (such as the age of, say, a passenger) as continuous, but at runtime the vehicle will face a finite number of options. We refer to these generalized models as *permutation processes*.

In Section 4.3, we focus on developing a theory of aggregation of permutation processes, which is crucial for Step IV. Specifically, we assume that societal preferences are represented as a single permutation process. Given a (finite) subset of alternatives, the permutation process induces a distribution over rankings of these alternatives. In the spirit of *distributional rank aggregation* [PPR15], we view this distribution over rankings as an *anonymous preference profile*, where the probability of a ranking is the fraction of voters whose preferences are represented by that ranking. This means we can apply a voting rule in order to aggregate the preferences — but *which* voting rule should we apply? And how can we compute the outcome *efficiently*? These are some of the central questions in computational social choice, but we show that in our context, under rather weak assumptions on the voting rule and permutation process, they are both moot, in the sense that it is easy to identify alternatives chosen by any "reasonable" voting rule. In slightly more detail, we define the notion of *swap dominance* between alternatives in a preference profile, and show that if the permutation process satisfies a natural property with respect to swap dominance (standard permutation processes do), and the voting rule is *swap-dominance efficient* (all common voting rules are), then any alternative that swap dominates all other alternatives is an acceptable outcome.

Armed with these theoretical developments, our task can be reduced to: learning a permutation process for each voter (Step II); summarizing these individual processes into a single permutation process that satisfies the required swap-dominance property (Step III); and using any swap-dominance efficient voting rule, which is computationally efficient given the swap-dominance property (Step IV).

In Section 4.4, we present a concrete algorithm that instantiates our approach, for a specific permutation process, namely the Thurstone-Mosteller (TM) Process [Thu27; Mos51], and with a specific linear parametrization of its underlying utility process in terms of the alternative features. While these simple choices have been made to illustrate the

framework, we note that, in principle, the framework can be instantiated with more general and complex permutation processes.

Finally, in Section 4.5, we implement and evaluate our algorithm. We first present simulation results from synthetic data that validate the accuracy of its learning and summarization components. More importantly, we implement our algorithm on the aforementioned Moral Machine dataset, and empirically evaluate the resultant system for choosing among alternatives in the autonomous vehicle domain. Taken together, these results suggest that our approach, and the algorithmic instantiation thereof, provide a computationally and statistically attractive method for automating decision making.

4.2 Preliminaries

Let $\mathcal{X}$ denote a potentially infinite set of alternatives. Given a finite subset $A \subseteq \mathcal{X}$, we are interested in the set $\mathcal{S}_A$ of *rankings* over A. Such a ranking $\sigma \in \mathcal{S}_A$ can be interpreted as mapping alternatives to their positions, i.e., $\sigma(a)$ is the position of $a \in A$ (smaller is more preferred). Let $a \succ_\sigma b$ denote that a is preferred to b in σ, that is, $\sigma(a) < \sigma(b)$. For $\sigma \in \mathcal{S}_A$ and $B \subseteq A$, let $\sigma|_B$ denote the ranking σ restricted to B. And for a distribution P over $\mathcal{S}_A$ and $B \subseteq A$, define $P|_B$ in the natural way to be the restriction of P to B, i.e., for all $\sigma' \in \mathcal{S}_B$,

$$P|_B(\sigma') = \sum_{\sigma \in \mathcal{S}_A:\ \sigma|_B = \sigma'} P(\sigma).$$

A *permutation process* $\{\Pi(A) : A \subseteq \mathcal{X}, |A| \in \mathbb{N}\}$ is a collection of distributions over $\mathcal{S}_A$ for every finite subset of alternatives A. We say that a permutation process is *consistent* if $\Pi(A)|_B = \Pi(B)$ for any finite subsets of alternatives $B \subseteq A \subseteq \mathcal{X}$. In other words, for a consistent permutation process Π, the distribution induced by Π over rankings of the alternatives in B is nothing but the distribution obtained by marginalizing out the extra alternatives $A \setminus B$ from the distribution induced by Π over rankings of the alternatives in A. This definition of consistency is closely related to the Luce Choice Axiom [Luc59].

A simple adaptation of folklore results [Mar95] shows that any permutation process that is consistent has a natural interpretation in terms of utilities. Specifically (and somewhat informally, to avoid introducing notation that will not be used later), given any consistent permutation process Π over a set of alternatives $\mathcal{X}$ (such that $|\mathcal{X}| \leq \aleph_1$), there exists a stochastic process U (indexed by $\mathcal{X}$) such that for any $A = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$, the probability of drawing $\sigma \in \mathcal{S}_A$ from $\Pi(A)$ is equal to the probability that $\text{sort}(U_{x_1}, U_{x_2}, \cdots, U_{x_m}) = \sigma$, where (perhaps obviously) $\text{sort}(\cdot)$ sorts the utilities in non-increasing order. We can allow ties in utilities, as long as $\text{sort}(\cdot)$ is endowed with some tie-breaking scheme, e.g., ties are broken lexicographically, which we will assume in the sequel. We refer to the stochastic process corresponding to a consistent permutation process as its *utility process*, since it is semantically meaningful to obtain a permutation via sorting by utility.

As examples of natural permutation processes, we adapt the definitions of two well-known *random utility models*. The (relatively minor) difference is that random utility models define a distribution over rankings over a fixed, finite subset of alternatives, whereas

permutation processes define a distribution for each finite subset of alternatives, given a potentially infinite space of alternatives.

- **Thurstone-Mosteller (TM) Process** [Thu27; Mos51]. A Thurstone-Mosteller Process (adaptation of Thurstones Case V model) is a consistent permutation process, whose utility process U is a Gaussian process with independent utilities and identical variances. In more detail, given a finite set of alternatives $\{x_1, x_2, \cdots, x_m\}$, the utilities $(U_{x_1}, U_{x_2}, \cdots, U_{x_m})$ are independent, and $U_{x_i} \sim \mathcal{N}(\mu_{x_i}, \frac{1}{2})$, where μ_{x_i} denotes the mode utility of alternative x_i.
- **Plackett-Luce (PL) Process** [Pla75; Luc59]. A Plackett-Luce Process is a consistent permutation process with the following utility process U: Given a finite set of alternatives $\{x_1, x_2, \cdots, x_m\}$, the utilities $(U_{x_1}, U_{x_2}, \cdots, U_{x_m})$ are independent, and each U_{x_i} has a Gumbel distribution with identical scale, i.e. $U_{x_i} \sim \mathcal{G}(\mu_{x_i}, \gamma)$, where $\mathcal{G}$ denotes the Gumbel distribution, and μ_{x_i} denotes the mode utility of alternative x_i. We note that Caron and Teh [CT12] consider a further Bayesian extension of the above PL process, with a Gamma process prior over the mode utility parameters.

4.3 Aggregation of Permutation Processes

In social choice theory, a *preference profile* is typically defined as a collection $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ of N rankings over a finite set of alternatives A, where σ_i represents the preferences of voter i. However, when the identity of voters does not play a role, we can instead talk about an *anonymous preference profile* $\pi \in [0,1]^{|A|!}$, where, for each $\sigma \in \mathcal{S}_A$, $\pi(\sigma) \in [0,1]$ is the *fraction* of voters whose preferences are represented by the ranking σ. Equivalently, it is the probability that a voter drawn uniformly at random from the population has the ranking σ.

How is this related to permutation processes? Given a permutation process Π and a finite subset $A \subseteq \mathcal{X}$, the distribution $\Pi(A)$ over rankings of A can be seen as an anonymous preference profile π, where for $\sigma \in \mathcal{S}_A$, $\pi(\sigma)$ is the probability of σ in $\Pi(A)$. As we shall see in Section 4.4, Step II (learning) gives us a permutation process for each voter, where $\pi(\sigma)$ represents our *confidence* that the preferences of the voter over A coincide with σ; and after Step III (summarization), we obtain a single permutation process that represents societal preferences.

Our focus in this section is the aggregation of anonymous preference profiles induced by a permutation process (Step IV), that is, the task of choosing the winning alternative(s). To this end, let us define an *anonymous social choice correspondence (SCC)* as a function f that maps any anonymous preference profile π over any finite and nonempty subset $A \subseteq \mathcal{X}$ to a nonempty subset of A. For example, under the ubiquitous *plurality* correspondence, the set of selected alternatives consists of alternatives with maximum first-place votes, i.e., $\arg\max_{a \in A} \sum_{\sigma \in \mathcal{S}_A : \sigma(a)=1} \pi(\sigma)$; and under the *Borda count* correspondence, denoting $|A| = m$, each vote awards $m - j$ points to the alternative ranked in position j, that is, the set of selected alternatives is $\arg\max_{a \in A} \sum_{j=1}^{m} (m-j) \sum_{\sigma \in \mathcal{S}_A : \sigma(a)=j} \pi(\sigma)$. We work with social choice *correspondences* instead of social choice *functions*, which return a single alternative in A, in order to smoothly handle ties.

4.3.1 Efficient Aggregation

Our main goal in this section is to address two related challenges. First, which (anonymous) social choice correspondence should we apply? There are many well-studied options, which satisfy different social choice axioms, and, in many cases, lead to completely different outcomes on the same preference profile. Second, how can we apply it in a computationally efficient way? This is not an easy task because, in general, we would need to explicitly construct the whole anonymous preference profile $\Pi(A)$, and then apply the SCC to it. The profile $\Pi(A)$ is of size $|A|!$, and hence this approach is intractable for a large $|A|$. Moreover, in some cases (such as the TM process), even computing the probability of a single ranking may be hard. The machinery we develop below allows us to completely circumvent these obstacles.

Since stating our general main result requires some setup, we first state a simpler instantiation of the result for the specific TM and PL permutation processes (we will directly use this instantiation in Section 4.4). Before doing so, we recall a few classic social choice axioms. We say that an anonymous SCC f is *monotonic* if the following conditions hold:

1. If $a \in f(\pi)$, and π' is obtained by pushing a upwards in the rankings, then $a \in f(\pi')$.
2. If $a \in f(\pi)$ and $b \notin f(\pi)$, and π' is obtained by pushing a upwards in the rankings, then $b \notin f(\pi')$.

In addition, an anonymous SCC is *neutral* if $f(\tau(\pi)) = \tau(f(\pi))$ for any anonymous preference profile π, and any permutation τ on the alternatives; that is, the SCC is symmetric with respect to the alternatives (in the same way that anonymity can be interpreted as symmetry with respect to voters).

Theorem 4.3.1. *Let Π be the TM or PL process, let $A \subseteq \mathcal{X}$ be a nonempty, finite subset of alternatives, and let $a \in \arg\max_{x \in A} \mu_x$. Moreover, let f be an anonymous SCC that is monotonic and neutral. Then $a \in f(\Pi(A))$.*

To understand the implications of the theorem, we first note that many of the common voting rules, including plurality, Borda count (and, in fact, all positional scoring rules), Copeland, maximin, and Bucklin (see, e.g., Brandt, Conitzer, Endriss, Lang, and Procaccia 2016), are associated with anonymous, neutral, and monotonic SCCs. Specifically, all of these rules have a notion of score, and the SCC simply selects all the alternatives tied for the top score (typically there is only one).[4] The theorem then implies that all of these rules would agree that, given a subset of alternatives A, an alternative $a \in A$ with maximum mode utility is an acceptable winner, i.e., it is at least tied for the highest score, if it is not the unique winner. As we will see in Section 4.4, such an alternative is very easy to identify, which is why, in our view, Theorem 4.3.1 gives a satisfying solution to the challenges posed at the beginning of this subsection. We emphasize that this is merely an instantiation of Theorem 4.3.7, which provides our result for general permutation processes.

The rest of this subsection is devoted to building the conceptual framework, and stating

[4]Readers who are experts in social choice have probably noted that there are no social choice *functions* that are both anonymous and neutral [Mou83], intuitively because it is impossible to break ties in a neutral way. This is precisely why we work with social choice *correspondences*.

the lemmas, required for the proof of Theorem 4.3.1, as well as the statement and proof of Theorem 4.3.7. We relegate all proofs to Appendix C.1.

Starting off, let π denote an anonymous preference profile (or distribution over rankings) over alternatives A. We define the ranking σ^{ab} as the ranking σ with alternatives a and b swapped, i.e. $\sigma^{ab}(x) = \sigma(x)$ if $x \in A \setminus \{a, b\}$, $\sigma^{ab}(b) = \sigma(a)$, and $\sigma^{ab}(a) = \sigma(b)$.

Definition 4.3.2. We say that alternative $a \in A$ *swap-dominates* alternative $b \in A$ in anonymous preference profile π over A — denoted by $a \triangleright_\pi b$ — if for every ranking $\sigma \in \mathcal{S}_A$ with $a \succ_\sigma b$ it holds that $\pi(\sigma) \geq \pi(\sigma^{ab})$.

In words, a swap-dominates b if every ranking that places a above b has at least as much weight as the ranking obtained by swapping the positions of a and b, and keeping everything else fixed. This is a very strong dominance relation, and, in particular, implies existing dominance notions such as *position dominance* [CPS16]. Next we define a property of social choice correspondences, which intuitively requires that the correspondence adhere to swap dominance relations, if they exist in a given anonymous preference profile.

Definition 4.3.3. An anonymous SCC f is said to be *swap-dominance-efficient (SwD-efficient)* if for every anonymous preference profile π and any two alternatives a and b, if a swap-dominates b in π, then $b \in f(\pi)$ implies $a \in f(\pi)$.

Because swap-dominance is such a strong dominance relation, SwD-efficiency is a very weak requirement, which is intuitively satisfied by almost any "reasonable" voting rule. This intuition is formalized in the following lemma.

Lemma 4.3.4. *Any anonymous SCC that satisfies monotonicity and neutrality is SwD-efficient.*

So far, we have defined a property, SwD-efficiency, that any SCC might potentially satisfy. But why is this useful in the context of aggregating permutation processes? We answer this question in Theorem 4.3.7, but before stating it, we need to introduce the definition of a property that a *permutation process* might satisfy.

Definition 4.3.5. Alternative $a \in \mathcal{X}$ swap-dominates alternative $b \in \mathcal{X}$ in the permutation process Π — denoted by $a \triangleright_\Pi b$ — if for every finite set of alternatives $A \subseteq \mathcal{X}$ such that $\{a, b\} \subseteq A$, a swap-dominates b in the anonymous preference profile $\Pi(A)$.

We recall that a *total preorder* is a binary relation that is transitive and total (and therefore reflexive).

Definition 4.3.6. A permutation process Π over $\mathcal{X}$ is said to be *SwD-compatible* if the binary relation $\triangleright_\Pi$ is a total preorder on $\mathcal{X}$.

We are now ready to state our main theorem.

Theorem 4.3.7. *Let f be an SwD-efficient anonymous SCC, and let Π be an SwD-compatible permutation process. Then for any finite subset of alternatives A, there exists $a \in A$ such that $a \triangleright_\Pi b$ for all $b \in A$. Moreover, $a \in f(\Pi(A))$.*

This theorem asserts that for any SwD-compatible permutation process, any SwD-efficient SCC (which, as noted above, include most natural SCCs, namely those that are monotonic and neutral), given any finite set of alternatives, will always select a very natural winner that swap-dominates other alternatives. A practical use of this theorem requires two things: to show that the permutation process is SwD-compatible, and that it is computationally tractable to select an alternative that swap-dominates other alternatives in

a finite subset. The next few lemmas provide some general recipes for establishing these properties for general permutation processes, and, in particular, we show that they indeed hold under the TM and PL processes. First, we have the following definition.

Definition 4.3.8. Alternative $a \in \mathcal{X}$ *dominates* alternative $b \in \mathcal{X}$ in utility process U if for every finite subset of alternatives containing a and b, $\{a, b, x_3, \dots x_m\} \subseteq \mathcal{X}$, and every vector of utilities $(u_1, u_2, u_3 \dots u_m) \in \mathbb{R}^m$ with $u_1 \geq u_2$, it holds that

$$\begin{aligned} &p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}(u_1, u_2, u_3 \dots u_m) \\ &\quad \geq p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}(u_2, u_1, u_3 \dots u_m), \end{aligned} \tag{4.1}$$

where $p_{(U_a, U_b, U_{x_3}, \dots U_{x_m})}$ is the density function of the random vector $(U_a, U_b, U_{x_3}, \dots U_{x_m})$.

Building on this definition, Lemmas 4.3.9 and 4.3.10 directly imply that the TM and PL processes are SwD-compatible, and complete the proof of Theorem 4.3.1 (see Appendix C.1).

Lemma 4.3.9. *Let Π be a consistent permutation process, and let U be its corresponding utility process. If alternative a dominates alternative b in U, then a swap-dominates b in Π.*

Lemma 4.3.10. *Under the TM and PL processes, alternative a dominates alternative b in the corresponding utility process if and only if $\mu_a \geq \mu_b$.*

4.3.2 Stability

It turns out that the machinery developed for the proof of Theorem 4.3.1 can be leveraged to establish an additional desirable property.

Definition 4.3.11. Given an anonymous SCC f, and a permutation process Π over $\mathcal{X}$, we say that the pair (Π, f) is *stable* if for any nonempty and finite subset of alternatives $A \subseteq \mathcal{X}$, and any nonempty subset $B \subseteq A$, $f(\Pi(A)) \cap B = f(\Pi(B))$ whenever $f(\Pi(A)) \cap B \neq \phi$.

Intuitively, stability means that applying f under the assumption that the set of alternatives is A, and then reducing to its subset B, is the same as directly reducing to B and then applying f. This notion is related to classic axioms studied by Sen [Sen71], specifically his *expansion* and *contraction* properties. In our setting, stability seems especially desirable, as our algorithm would potentially face decisions over many different subsets of alternatives, and the absence of stability may lead to glaringly inconsistent choices.

Theorem 4.3.12. *Let Π be the TM or PL process, and let f be the Borda count or Copeland SCC. Then the pair (Π, f) is stable.*

The definition of the Copeland SCC, and the proof of the theorem, are relegated to Appendix C.2. Among other things, the proof requires a stronger notion of SwD-efficiency, which, as we show, is satisfied by Borda count and Copeland, and potentially by other appealing SCCs.

4.4 Instantiation of Our Approach

In this section, we instantiate our approach for ethical decision making, as outlined in Section 4.1. In order to present a concrete algorithm, we consider a specific permutation

process, namely the TM process with a linear parameterization of the utility process parameters as a function of the alternative features.

Let the set of alternatives be given by $\mathcal{X} \subseteq \mathbb{R}^d$, i.e. each alternative is represented by a vector of d features. Furthermore, let N denote the total number of voters. Assume for now that the data-collection step (Step I) is complete, i.e., we have some pairwise comparisons for each voter; we will revisit this step in Section 4.5.

Step II: Learning. For each voter, we learn a TM process using his pairwise comparisons to represent his preferences. We assume that the mode utility of an alternative x depends linearly on its features, i.e., $\mu_x = \boldsymbol{\beta}^\intercal x$. Note that we do not need an intercept term, since we care only about the relative ordering of utilities. Also note that the parameter $\boldsymbol{\beta} \in \mathbb{R}^d$ completely describes the TM process, and hence the parameters $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots \boldsymbol{\beta}_N$ completely describe the models of all voters.

Next we provide a computationally efficient method for learning the parameter $\boldsymbol{\beta}$ for a particular voter. Let $(X_1, Z_1), (X_2, Z_2), \cdots, (X_n, Z_n)$ denote the pairwise comparison data of the voter. Specifically, the ordered pair (X_j, Z_j) denotes the j^{th} pair of alternatives compared by the voter, and the fact that the voter chose X_j over Z_j. We use maximum likelihood estimation to estimate $\boldsymbol{\beta}$. The log-likelihood function is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta}) &= \log \left[\prod_{j=1}^{n} P(X_j \succ Z_j; \boldsymbol{\beta}) \right] \\
&= \sum_{j=1}^{n} \log P(U_{X_j} > U_{Z_j}; \boldsymbol{\beta}) \\
&= \sum_{j=1}^{n} \log \Phi\left(\boldsymbol{\beta}^\intercal (X_j - Z_j)\right),
\end{aligned}$$

where Φ is the cumulative distribution function of the standard normal distribution, and the last transition holds because $U_x \sim \mathcal{N}(\boldsymbol{\beta}^\intercal x, \frac{1}{2})$. Note that the standard normal CDF Φ is a log-concave function. This makes the log-likelihood concave in $\boldsymbol{\beta}$, hence we can maximize it efficiently.

Step III: Summarization. After completing Step II, we have N TM processes represented by the parameters $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots \boldsymbol{\beta}_N$. In Step III, we bundle these individual models into a single permutation process $\hat{\Pi}$, which, in the current instantiation, is also a TM process with parameter $\hat{\boldsymbol{\beta}}$ (see Section 4.6 for a discussion of this point). We perform this step because we must be able to make decisions *fast*, in Step IV. For example, in the autonomous vehicle domain, the AI would only have a split second to make a decision in case of emergency; aggregating information from millions of voters *in real time* will not do. By contrast, Step III is performed offline, and provides the basis for fast aggregation.

Let $\Pi^{\boldsymbol{\beta}}$ denote the TM process with parameter $\boldsymbol{\beta}$. Given a finite subset of alternatives $A \subseteq \mathcal{X}$, the anonymous preference profile generated by the model of voter i is given by $\Pi^{\boldsymbol{\beta}_i}(A)$. Ideally, we would like the summary model to be such that the profile generated by it, $\hat{\Pi}(A)$, is as close as possible to $\Pi^*(A) = \frac{1}{N} \sum_{i=1}^{N} \Pi^{\boldsymbol{\beta}_i}(A)$, the mean profile obtained by

giving equal importance to each voter. However, there does not appear to be a straightforward method to compute the "best" $\hat{\boldsymbol{\beta}}$, since the profiles generated by the TM processes do not have an explicit form. Hence, we use utilities as a proxy for the quality of $\hat{\boldsymbol{\beta}}$. Specifically, we find $\hat{\boldsymbol{\beta}}$ such that the summary model induces utilities that are as close as possible to the mean of the utilities induced by the per-voter models, i.e., we want $U_x^{\hat{\boldsymbol{\beta}}}$ to be as close as possible (in terms of KL divergence) to $\frac{1}{N}\sum_{i=1}^{N} U_x^{\boldsymbol{\beta_i}}$ for each $x \in \mathcal{X}$, where $U_x^{\boldsymbol{\beta}}$ denotes the utility of x under TM process with parameter $\boldsymbol{\beta}$. This is achieved by taking $\hat{\boldsymbol{\beta}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$, as shown by the following proposition (whose proof appears in Appendix C.3).

Proposition 4.4.1. *The vector* $\boldsymbol{\beta} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$ *minimizes* $KL\left(\frac{1}{N}\sum_{i=1}^{N} U_x^{\boldsymbol{\beta}_i} \middle\| U_x^{\boldsymbol{\beta}}\right)$ *for any* $x \in \mathcal{X}$.

Step IV: Aggregation. As a result of Step III, we have exactly one (summary) TM process $\hat{\Pi}$ (with parameter $\hat{\boldsymbol{\beta}} = \bar{\boldsymbol{\beta}}$) to work with at runtime. Given a finite set of alternatives $A = \{x_1, x_2, \cdots, x_m\}$, we must aggregate the preferences represented by the anonymous preference profile $\hat{\Pi}(A)$. This is where the machinery of Section 4.3 comes in: We simply need to select an alternative that has maximum mode utility among $\hat{\boldsymbol{\beta}}^\intercal x_1, \hat{\boldsymbol{\beta}}^\intercal x_2, \cdots, \hat{\boldsymbol{\beta}}^\intercal x_m$. Such an alternative would be selected by any anonymous SCC that is monotonic and neutral, when applied to $\hat{\Pi}(A)$, as shown by Theorem 4.3.1. Moreover, this aggregation method is equivalent to applying the Borda count or Copeland SCCs (due to Lemmas C.2.5, C.2.7). Hence, we also have the desired stability property, as shown by Theorem 4.3.12.

4.5 Implementation and Evaluation

In this section, we implement the algorithm presented in Section 4.4, and empirically evaluate it. We start with an implementation on synthetic data, which allows us to effectively validate both Steps II and III of our approach. We then describe the Moral Machine dataset mentioned in Section 4.1, present the implementation of our algorithm on this dataset, and evaluate the resultant system for ethical decision making in the autonomous vehicle domain (focusing on Step III).

4.5.1 Synthetic Data

Setup. We represent the preferences of each voter using a TM process. Let $\boldsymbol{\beta}_i$ denote the true parameter corresponding to the model of voter i. We sample $\boldsymbol{\beta}_i$ from $\mathcal{N}(\mathbf{m}, I_d)$ (independently for each voter i), where each mean m_j is sampled independently from the uniform distribution $\mathcal{U}(-1, 1)$, and the number of features is $d = 10$.

In each instance (defined by a subset of alternatives A with $|A| = 5$), the desired winner is given by the application of Borda count to the mean of the profiles of the voters. In more detail, we compute the anonymous preference profile of each voter $\Pi^{\boldsymbol{\beta}_i}(A)$, and then take a mean across all the voters to obtain the desired profile $\frac{1}{N}\sum_{i=1}^{N} \Pi^{\boldsymbol{\beta}_i}(A)$. We then apply Borda count to this profile to obtain the winner. Note that, since we are dealing with TM processes, we cannot explicitly construct $\Pi^{\boldsymbol{\beta}_i}(A)$; we therefore estimate it by sampling rankings according to the TM process of voter i.

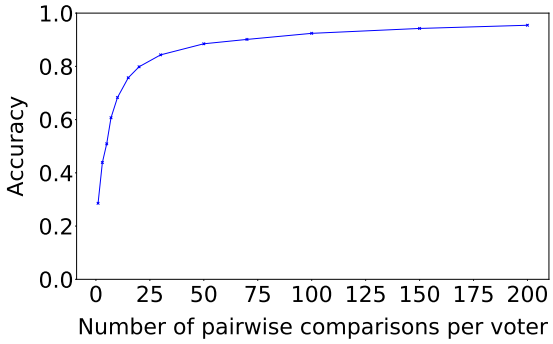


Figure 4.1: Accuracy of Step II (synthetic data)

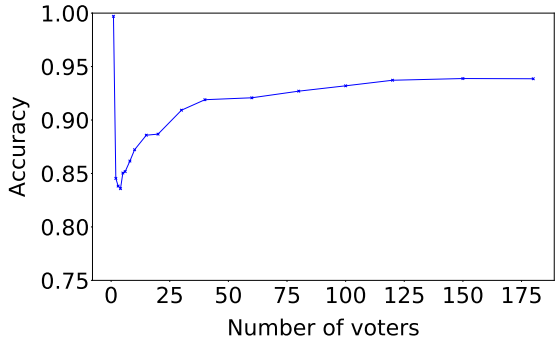


Figure 4.2: Accuracy of Step III (synthetic data)

Evaluation of Step II (Learning). In practice, the algorithm does not have access to the true parameter $\boldsymbol{\beta}_i$ of voter i, but only to pairwise comparisons, from which we learn the parameters. Thus we compare the computation of the winner (following the approach described above) using the true parameters, and using the learned parameters as in Step II. We report the accuracy as the fraction of instances, out of 100 test instances, in which the two outcomes match.

To generate each pairwise comparison of voter i, for each of $N = 20$ voters, we first sample two alternatives x_1 and x_2 independently from $\mathcal{N}(\mathbf{0}, I_d)$. Then, we sample their utilities U_{x_1} and U_{x_2} from $\mathcal{N}(\boldsymbol{\beta}_i^\intercal x_1, \frac{1}{2})$ and $\mathcal{N}(\boldsymbol{\beta}_i^\intercal x_2, \frac{1}{2})$, respectively. Of course, the voter prefers the alternative with higher sampled utility. Once we have the comparisons, we learn the parameter $\boldsymbol{\beta}_i$ by computing the MLE (as explained in Step II of Section 4.4). In our results, we vary the number of pairwise comparisons per voter and compute the accuracy to obtain the learning curve shown in Figure 4.1. Each datapoint in the graph is averaged over 50 runs. Observe that the accuracy quickly increases as the number of pairwise comparisons increases, and with just 30 pairwise comparisons we achieve an accuracy of 84.3%. With 100 pairwise comparisons, the accuracy is 92.4%.

Evaluation of Step III (Summarization). To evaluate Step III, we assume that we have access to the true parameters $\boldsymbol{\beta}_i$, and wish to determine the accuracy loss incurred in the summarization step, where we summarize the individual TM models into a single TM model. As described in Section 4.4, we compute $\bar{\boldsymbol{\beta}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\beta}_i$, and, given a subset A (which again has cardinality 5), we aggregate using Step IV, since we now have just one TM process. For each instance, we contrast our computed winner with the desired winner as computed previously. We vary the number of voters and compute the accuracy to obtain Figure 4.2. The accuracies are averaged over 50 runs. Observe that the accuracy increases to 93.9% as the number of voters increases. In practice we expect to have access to thousands, even millions, of votes (see Section 4.5.2). We conclude that, surprisingly, the expected loss in accuracy due to summarization is quite small.

Robustness. Our results are robust to the choice of parameters, as we demonstrate in Appendix C.4.

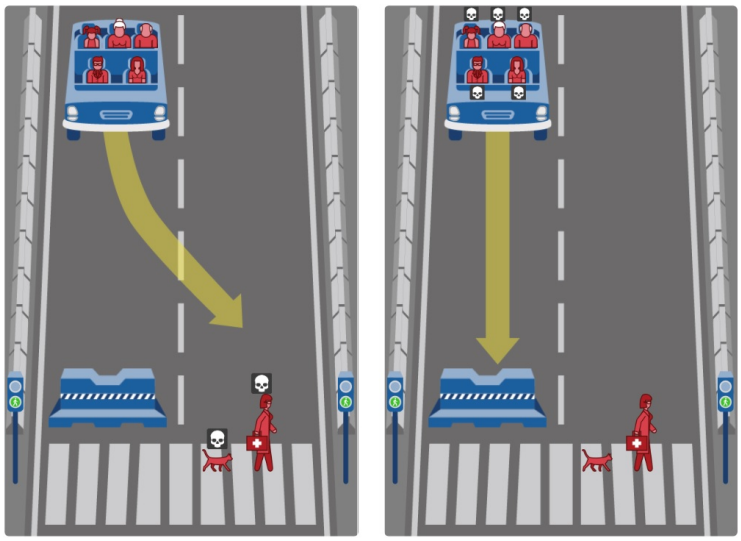


Figure 4.3: *Moral Machine* — Judge interface [Awa+18]. This particular choice is between a group of pedestrians that includes a female doctor and a cat crossing on a green light, and a group of passengers including a woman, a male executive, an elderly man, an elderly woman, and a girl.

4.5.2 Moral Machine Data

As mentioned in Section 4.1, we apply our approach on the Moral Machine dataset only as a proof of concept. Moral Machine is a platform for gathering data on human perception of the moral acceptability of decisions made by autonomous vehicles faced with choosing which humans to harm and which to save [Awa+18]. The main interface of Moral Machine is the Judge mode. This interface generates sessions of random moral dilemmas. In each session, a user is faced with 13 instances. Each instance features an autonomous vehicle with a brake failure, facing a moral dilemma with two possible alternatives, that is, each instance is a pairwise comparison. Each of the two alternatives corresponds to sacrificing the lives of one group of characters to spare those of another group of characters. Figure 4.3 shows an example of such an instance. Respondents choose the outcome that they prefer the autonomous vehicle to make.

Each alternative is characterized by 22 features: relation to the autonomous vehicle (passengers or pedestrians), legality (no legality, explicitly legal crossing, or explicitly illegal crossing), and counts of 20 character types, including ones like man, woman, pregnant woman, male athlete, female doctor, dog, etc. When sampling from the 20 characters, some instances are generated to have an easy-to-interpret tradeoff with respect to some dimension, such as gender (males on one side vs. females on the other), age (elderly vs. young), fitness (large vs. fit), etc., while other instances have groups consisting of completely randomized characters being sacrificed in either alternative. Alternatives with all possible combinations of these features are considered, except for the legality feature in cases when passengers are sacrificed. In addition, each alternative has a derived feature, "number of characters," which is simply the sum of counts of the 20 character types (making $d = 23$).

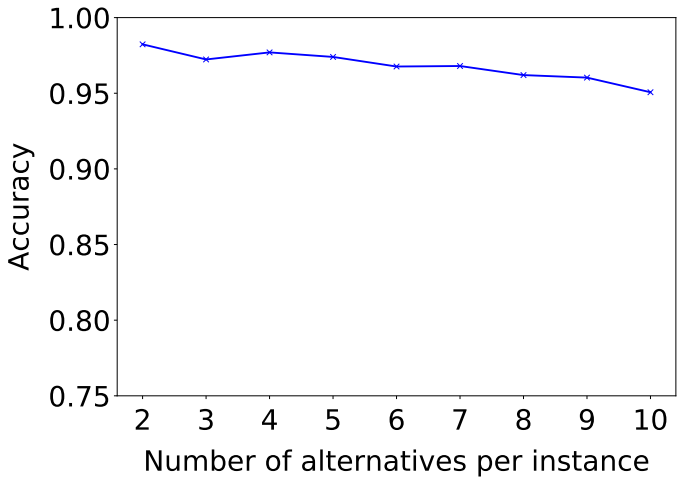


Figure 4.4: Accuracy of Step III (Moral Machine data)

As mentioned in Section 4.1, the Moral Machine dataset consists of preference data from 1,303,778 voters, amounting to a total of 18,254,285 pairwise comparisons. We used this dataset to learn the $\boldsymbol{\beta}$ parameters of all 1.3 million voters (Step II, as given in Section 4.4). Next, we took the mean of all of these $\boldsymbol{\beta}$ vectors to obtain $\hat{\boldsymbol{\beta}}$ (Step III). This gave us an implemented system, which can be used to make real-time choices between any finite subset of alternatives.

Importantly, the methodology we used, in Section 4.5.1, to evaluate Step II on the synthetic data cannot be applied to the Moral Machine data, because we do not know which alternative would be selected by aggregating the preferences of the actual 1.3 million voters over a subset of alternatives. However, we can apply a methodology similar to that of Section 4.5.1 in order to evaluate Step III. Specifically, as in Section 4.5.1, we wish to compare the winner obtained using the summarized model, with the winner obtained by applying Borda count to the mean of the anonymous preference profiles of the voters.

An obstacle is that now we have a total of 1.3 million voters, and hence it would take an extremely long time to calculate the anonymous preference profile of each voter and take their mean (this was the motivation for having Step III in the first place). So, instead, we estimate the mean profile by sampling rankings, i.e., we sample a voter i uniformly at random, and then sample a ranking from the TM process of voter i; such a sampled ranking is an i.i.d. sample from the mean anonymous profile. Then, we apply Borda count as before to obtain the desired winner (note that this approach is still too expensive to use in real time). The winner according to the summarized model is computed exactly as before, and is just as efficient even with 1.3 million voters.

Using this methodology, we computed accuracy on 3000 test instances, i.e., the fraction of instances in which the two winners match. Figure 4.4 shows the results as the number of alternatives per instance is increased from 2 to 10. Observe that the accuracy is as high as 98.2% at 2 alternatives per instance, and gracefully degrades to 95.1% at 10.

4.6 Discussion

As mentioned in Section 4.4, we have made some specific choices to instantiate our approach. We discuss two of the most consequential choices.

First, we assume that the mode utilities have a linear structure. This means that, under the TM model, the estimation of the maximum likelihood parameters is a convex program (see Section 4.4), hence we can learn the preferences of millions of voters, as in the Moral Machine dataset. Moreover, a straightforward summarization method works well. However, dealing with a richer representation for utilities would require new methods for both learning and summarization (Steps II and III).

Second, the instantiation given in Section 4.4 summarizes the N individual TM models as a single TM model. While the empirical results of Section 4.5 suggest that this method is quite accurate, even higher accuracy can potentially be achieved by summarizing the N models as a *mixture* of K models, for a relatively small K. This leads to two technical challenges: What is a good algorithm for generating this mixture of, say, TM models? And, since the framework of Section 4.3 would not apply, how should such a mixture be aggregated — does the (apparently mild) increase in accuracy come at great cost to computational efficiency?

Finally, we would like to emphasize that our approach has been applied to the Moral Machine dataset mainly as a proof of concept, and we are not advocating the use of this approach as is for this domain in real life. We believe that the implementation of our algorithm on the Moral Machine dataset has yielded a system which, arguably, can make *credible* decisions on ethical dilemmas in the autonomous vehicle domain (when all other options have failed), but this work is clearly not the end-all solution. To be usable in the real world, the framework would have to be extended to incorporate ethical and legal principles that come in to play. Further, a more careful study of ethics is needed to confirm whether this approach is justifiable for such a domain — as people's lives are on the line, and we do not want mob rule to come into play.

For a real world application and deployment of this approach, we would like to point the reader to follow up work by Lee et al. [Lee+19]. The goal in this work is to design and deploy an algorithm that would assist a food bank in automating decisions they most frequently face: given an incoming food donation, which recipient organization (such as a housing authority or food pantry) should receive it? The voters in the implementation are stakeholders: donors, recipients, volunteers (who pick up the food from the donor and deliver it to the recipient), and employees. They have collected roughly 100 pairwise comparisons from each voter, and apply the virtual democracy framework to automate decisions. This approach has been deployed to assist 412 Food Rescue, a nonprofit in Pittsburgh, PA, that aims to fight food waste, and received highly positive reception and feedback from the several stakeholders involved. Finally, for technical problems that arise when trying to use this framework for such a domain, we would like to point the reader to another follow up work by Kahng, Lee, Noothigattu, Procaccia, and Psomas [Kah+19].

Chapter 5

Loss Functions, Axioms, and Peer Review

It is common to see a handful of reviewers reject a highly novel paper, because they view, say, extensive experiments as far more important than novelty, whereas the community as a whole would have embraced the paper. More generally, the disparate mapping of criteria scores to final recommendations by different reviewers is a major source of inconsistency in peer review. In this chapter, we present a framework inspired by empirical risk minimization (ERM) for learning the community's aggregate mapping. The key challenge that arises is the specification of a loss function for ERM. We consider the class of $L(p, q)$ loss functions, which is a matrix-extension of the standard class of L_p losses on vectors; here the choice of the loss function amounts to choosing the hyperparameters $p, q \in [1, \infty]$. To deal with the absence of ground truth in our problem, we instead draw on computational social choice to identify desirable values of the hyperparameters p and q. Specifically, we characterize $p = q = 1$ as the only choice of these hyperparameters that satisfies three natural axiomatic properties. Finally, we implement and apply our approach to reviews from IJCAI 2017.

5.1 Introduction

The essence of science is the search for objective truth, yet scientific work is typically evaluated through peer review[1] — a notoriously subjective process [Chu05; Lam09; BMS87; HGC03; Mah77; KTP77]. One prominent source of subjectivity is the disparity across reviewers in terms of their emphasis on the various criteria used for the overall evaluation of a paper. Lee [Lee15] refers to this disparity as *commensuration bias*, and describes it as follows:

> *"In peer review, reviewers, editors, and grant program officers must make interpretive decisions about how to weight the relative importance of qualitatively different peer review criteria — such as novelty, significance, and methodological soundness — in their assessments of a submission's final/overall value. Not all peer review criteria get equal weight; further, weightings can vary across reviewers and contexts even when reviewers are given identical instructions."*

[1] Even papers about peer review are subject to peer review, the irony of which has not escaped us.

Lee [Lee15] further argues that commensuration bias "illuminates how intellectual priorities in individual peer review judgments can collectively subvert the attainment of community-wide goals" and that it "permits and facilitates problematic patterns of publication and funding in science." There have been, however, very few attempts to address this problem.

A fascinating exception, which serves as a case in point, is the 27th AAAI Conference on Artificial Intelligence (AAAI 2013). Reviewers were asked to score papers, on a scale of 1–6, according to the following criteria: technical quality, experimental analysis, formal analysis, clarity/presentation, novelty of the question, novelty of the solution, breadth of interest, and potential impact. The admirable goal of the program chairs was to select "exciting but imperfect papers" over "safe but solid" papers, and, to this end, they provided detailed instructions on how to map the foregoing criteria to an overall recommendation. For example, the preimage of 'strong accept' is "a 5 or 6 in some category, no 1 in any category," that is, reviewers were instructed to strongly accept a paper that has a 5 or 6 in, say, clarity, but is below average according to each and every other criterion (i.e., a clearly boring paper). It turns out that the handcrafted mapping did not work well, and many of the reviewers chose to not follow these instructions. Indeed, handcrafting such a mapping requires specifying an 8-dimensional function, which is quite a non-trivial task.[2] Consequently, in this paper we do away with a manual handcrafting approach to this problem.

Instead, we propose a data-driven approach based on machine learning, designed to learn a mapping from criteria scores to recommendations capturing the opinion of the entire (reviewer) community. From a machine learning perspective, the examples are reviews, each consisting of criteria scores (the input point) and an overall recommendation (the label). We make the innocuous assumption that each reviewer has a *monotonic* mapping in mind, in the sense that a paper whose scores are at least as high as those of another paper on every criterion would receive an overall recommendation that is at least as high; the reviews submitted by a particular reviewer can be seen as observations of that mapping. Given this data, our goal is to learn a *single monotonic mapping* that minimizes a loss function (which we discuss momentarily). We can then apply this mapping to the criteria scores associated with each review to obtain new overall recommendations, which replace the original ones.

Our approach to learn this mapping is inspired by empirical risk minimization (ERM). In more detail, for some loss function, our approach is to find a mapping that, among all monotonic mappings from criteria scores to the overall scores, minimizes the loss between its outputs and the overall scores given by reviewers across all reviews. However, the choice of loss function may significantly affect the final outcome, so that choice is a key issue.

Specifically, we focus on the family of $L(p, q)$ loss functions, with hyperparameters $p, q \in [1, \infty]$, which is a matrix-extension of the more popular family of L_p losses on vectors. Our question, then, is:

What values of the hyperparameters $p \in [1, \infty]$ and $q \in [1, \infty]$ in the specifica-

[2]See also [MA08] for a similar case in the peer-review process of the OSDI conference – OSDI 2006 did not allow reviewers to report an overall score, but instead, the PC co-chairs synthesized this score from a weighted combination of criteria scores. Here, we instead take a community-based approach and learn a mapping common to the community of reviewers. Further, we do not assume linear aggregation of the criteria scores, but allow a more general monotonic mapping.

tion of the $L(p, q)$ loss function should be used?

A challenge we must address is the absence of any ground truth in peer review. To this end, take the perspective of *computational social choice* [Bra+16], since our framework aggregates individual opinions over mappings into a consensus mapping. From this viewpoint, it is natural to select the loss function so that the resulting aggregation method satisfies socially desirable properties, such as *consensus* (if all reviewers agree then the aggregate mapping should coincide with their recommendations), *efficiency* (if one paper dominates another then its overall recommendation should be at least as high), and *strategyproofness* (reviewers cannot pull the aggregate mapping closer to their own recommendations by misreporting them).

With this background, the main contributions of this paper are as follows. We first provide a principled framework for addressing the issue of subjectivity regarding the various criteria in peer review.

Our main theoretical result is a characterization theorem that gives a decisive answer to the question of choosing the loss function for ERM: the three aforementioned properties are satisfied *if and only if* the hyperparameters are set as $p = q = 1$. This result singles out an instantiation of our approach that we view as particularly attractive and well grounded.

We also provide empirical results, which analyze properties of our approach when applied to a dataset of 9197 reviews from IJCAI 2017. One vignette is that the papers selected by $L(1, 1)$ aggregation have a 79.2% overlap with actual list of accepted papers, suggesting that our approach makes a significant difference compared to the status quo (arguably for the better).

Finally, we note that the approach taken in this paper may find other applications. Indeed, the problem of selecting a loss function is ubiquitous in machine learning [Ros+04; MV08; MBM18], and the axiomatic approach provides a novel way of addressing it. Going beyond loss functions, machine learning researchers frequently face the difficulty of picking an appropriate hypothesis class or values for certain hyperparameters.[3] Thus, in problem settings where such choices must be made — particularly in emerging applications of machine learning (such as peer review) — the use of natural axioms can help guide these choices.

5.2 Our Framework

Suppose there are n reviewers $\mathcal{R} = \{1, 2, \ldots, n\}$, and a set $\mathcal{P}$ of m papers, denoted using letters such as a, b, c. Each reviewer i reviews a subset of papers, denoted by $P(i) \subseteq \mathcal{P}$. Conversely, let $R(a)$ denote the set of all reviewers who review paper a. Each reviewer assigns scores to each of their papers on d different criteria, such as novelty, experimental analysis, and technical quality, and also gives an overall recommendation. We denote the *criteria scores* given by reviewer i to paper a by $\mathbf{x}_{ia}$, and the corresponding *overall recommendation* by y_{ia}. Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_d$ denote the domains of the d criteria scores, and let $\mathbb{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$. Also, let $\mathbb{Y}$ denote the domain of the overall recommendations.

[3]Popular techniques such as cross-validation for choosing hyperparameters also in turn depend on specification of a loss function.

For concreteness, we assume that each $\mathcal{X}_k$ as well as $\mathbb{Y}$ is the real line. However, our results hold more generally, even if these domains are non-singleton intervals in $\mathbb{R}$, for instance.

We further assume that each reviewer has a monotonic function in mind that they use to compute the overall recommendation for a paper from its criteria scores. By a monotonic function, we mean that given any two score vectors $\mathbf{x}$ and $\mathbf{x}'$, if $\mathbf{x}$ is greater than or equal to $\mathbf{x}'$ on all coordinates, then the function's value on $\mathbf{x}$ must be at least as high as its value on $\mathbf{x}'$. Formally, for each reviewer i, there exists $g_i^\star \in \mathcal{F}$ such that $y_{ia} = g_i^\star(\mathbf{x}_{ia})$ for all $a \in P(i)$, where

$$\mathcal{F} = \{f : \mathbb{X} \to \mathbb{Y} \mid \forall \mathbf{x}, \mathbf{x}' \in \mathbb{X}, \mathbf{x} \geq \mathbf{x}' \Rightarrow f(\mathbf{x}) \geq f(\mathbf{x}')\}$$

is the set of all monotonic functions.

5.2.1 Loss Functions

Recall that our goal is to use all criteria scores, and their corresponding overall recommendations, to learn an aggregate function $\widehat{f}$ that captures the opinions of all reviewers on how criteria scores should be mapped to recommendations. Inspired by empirical risk minimization, we do this by computing the function in $\mathcal{F}$ that minimizes the $L(p, q)$ loss on the data. In more detail, given hyperparameters $p, q \in [1, \infty]$, we compute

$$\widehat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} \left\{ \sum_{i \in \mathcal{R}} \left[\sum_{a \in P(i)} |y_{ia} - f(\mathbf{x}_{ia})|^p \right]^{\frac{q}{p}} \right\}^{\frac{1}{q}}. \tag{5.1}$$

In words, for a function f, the $L(p, q)$ loss is the L_q norm taken over the loss associated with individual reviewers, where the latter loss is defined as the L_p norm computed on the error of f with respect to the reviewer's overall recommendations. The $L(p, q)$ loss is a matrix-extension of the more popular L_p losses on vectors, and relates to the $L(p, q)$ norm of a matrix which has had many applications in machine learning [Din+06; KDH11; Nie+10]. We refer to aggregation by minimizing $L(p, q)$ loss as defined in Equation (5.1) as $L(p, q)$ *aggregation*.

Equation (5.1) does not specify how to break ties between multiple minimizers. For concreteness, we select the minimizer $\widehat{f}$ with minimum empirical L_2 norm. Formally, letting

$$\widehat{F} = \operatorname*{argmin}_{f \in \mathcal{F}} \left\{ \sum_{i \in \mathcal{R}} \left[\sum_{a \in P(i)} |y_{ia} - f(\mathbf{x}_{ia})|^p \right]^{\frac{q}{p}} \right\}^{\frac{1}{q}}$$

be the set of all $L(p, q)$ loss minimizers, we break ties by choosing

$$\widehat{f} \in \operatorname*{argmin}_{f \in \widehat{F}} \sqrt{\sum_{i \in \mathcal{R}} \sum_{a \in P(i)} f(\mathbf{x}_{ia})^2}. \tag{5.2}$$

Observe that since the $L(p, q)$ loss and constraint set are convex, $\widehat{F}$ is also a convex set. Hence, $\widehat{f}$ as defined by Equation (5.2) is unique. We emphasize that although we use

minimum L_2 norm for tie-breaking, all of our results hold under any reasonable tie-breaking method, such as the minimum L_k norm for any $k \in (1, \infty)$.

Once the function $\widehat{f}$ has been computed, it can be applied to every review (for all reviewers i and papers a) to obtain a new overall recommendation $\widehat{f}(\mathbf{x}_{ia})$. There is a separate — almost orthogonal — question of how to aggregate the overall recommendations of several reviewers on a paper into a single recommendation (typically this is done by taking the average). In our theoretical results we are agnostic to how this additional aggregation step is performed, but we return to it in our experiments in Section 5.4.

We remark that an alternative approach would be to learn a monotonic function $\widehat{g}_i : \mathbb{X} \to \mathbb{Y}$ for each reviewer (which best captures their recommendations), and then aggregate these functions into a single function $\widehat{f}$. We chose not to pursue this approach, because in practice there are very few examples per reviewer, so it is implausible that we would be able to accurately learn the reviewers' individual functions.

5.2.2 Axiomatic Properties

In social choice theory, the most common approach — primarily attributed to Arrow [Arr51] — for comparing different aggregation methods is to determine which desirable axioms they satisfy. We take the same approach in order to determine the values of the hyperparameters p and q for the $L(p, q)$ aggregation in Equation (5.1).

We stress that axioms are defined for aggregation methods and not aggregate functions. Informally, an aggregation method is a function that takes as input all the reviews $\{(\mathbf{x}_{ia}, y_{ia})\}_{i \in \mathcal{R}, a \in P(i)}$, and outputs an aggregate function $\widehat{f} : \mathbb{X} \to \mathbb{Y}$. We do not define an aggregation method formally to avoid introducing cumbersome notation that will largely be useless later. It is clear that for any choice of hyperparameters $p, q \in [1, \infty]$, $L(p, q)$ aggregation (with tie-breaking as defined by Equation 5.2) is an aggregation method.

Social choice theory essentially relies on counterfactual reasoning to identify scenarios where it is clear how an aggregation method should behave. To give one example, the *Pareto efficiency* property of voting rules states that if all voters prefer alternative a to alternative b, then b should *not* be elected; this situation is extremely unlikely to occur, yet Pareto efficiency is obviously a property that any reasonable voting must satisfy. With this principle in mind, we identify a setting in our problem where the requirements are very clear, and then define our axioms in that setting.

For all of our axioms, we restrict attention to scenarios where every reviewer reviews every paper, that is, $P(i) = \mathcal{P}$ for every i. Moreover, we assume that the papers have 'objective' criteria scores, that is, the criteria scores given to a paper are the same across all reviewers, so the only source of disagreement is how the criteria scores should be mapped to an overall recommendation. We can then denote the criteria scores of a paper a simply as $\mathbf{x}_a$, as opposed to $\mathbf{x}_{ia}$, since they do not depend on i. We stress that our framework does *not* require these assumptions to hold — they are only used in our axiomatic characterization, namely Theorem 5.3.1 in the next section.

An axiom is satisfied by an aggregation method if its statement holds for every possible number of reviewers n and number of papers m, and for all possible criteria scores and

overall recommendations. We start with the simplest axiom, consensus, which informally states that if there is a paper such that all reviewers give it the same overall recommendation, then $\widehat{f}$ must agree with the reviewers; this axiom is closely related to the *unanimity* axiom in social choice.

Axiom 5.2.1 (Consensus)**.** *For any paper $a \in \mathcal{P}$, if all reviewers report identical overall recommendations $y_{1a} = y_{2a} = \cdots = y_{ma} = r$ for some $r \in \mathbb{Y}$, then $\widehat{f}(\mathbf{x}_a) = r$.*

Before presenting the next axiom, we require another definition: we say that paper $a \in \mathcal{P}$ *dominates* paper $b \in \mathcal{P}$ if there exists a bijection $\sigma : \mathcal{R} \to \mathcal{R}$ such that for all $i \in \mathcal{R}$, $y_{ia} \geq y_{\sigma(i)b}$. Equivalently (and less formally), paper a dominates paper b if the *sorted* overall recommendations given to a pointwise-dominate the *sorted* overall recommendations given to b. Intuitively, in this situation, a should receive a (weakly) higher overall recommendation than b, which is exactly what the axiom requires; it is similar to the classic Pareto efficiency axiom mentioned above.

Axiom 5.2.2 (Efficiency)**.** *For any pair of papers $a, b \in \mathcal{P}$, if a dominates b, then $\widehat{f}(\mathbf{x}_a) \geq \widehat{f}(\mathbf{x}_b)$.*

Our positive result, which will be presented shortly, satisfies this notion of efficiency. On the other hand, we also use this axiom to prove a negative result; an important note is that the negative result requires a condition that is significantly weaker than the aforementioned definition of efficiency. We revisit this point at the end of Section 5.3.2.

Our final axiom is *strategyproofness*, a game-theoretic property that plays a major role in social choice theory [Mou83]. Intuitively, strategyproofness means that reviewers have no incentive to misreport their overall recommendations: They cannot bring the aggregate recommendations — the community's consensus about the relative importance of various criteria — closer to their own through strategic manipulation.

Axiom 5.2.3 (Strategyproofness)**.** *For each reviewer $i \in \mathcal{R}$, and all possible manipulated recommendations $\mathbf{y}'_i \in \mathbb{Y}^m$, if $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{im})$ is replaced with $\mathbf{y}'_i$, then*

$$\|(\widehat{f}(\mathbf{x}_1), \ldots, \widehat{f}(\mathbf{x}_m)) - \mathbf{y}_i\|_2 \leq \|(\widehat{g}(\mathbf{x}_1), \ldots, \widehat{g}(\mathbf{x}_m)) - \mathbf{y}_i\|_2, \tag{5.3}$$

where $\widehat{f}$ and $\widehat{g}$ are the aggregate functions obtained from the original and manipulated reviews, respectively.

The implicit 'utilities' in this axiom (5.3) are defined in terms of the L_2 norm. This choice is made only for concreteness, and all our results hold for any norm L_ℓ, $\ell \in [1, \infty]$, in the definition (5.3).

5.3 Main Result

In Section 5.2, we introduced $L(p, q)$ aggregation as a family of rules for aggregating individual opinions towards a consensus mapping from criteria scores to recommendations. But that definition, in and of itself, leaves open the question of how to choose the values of p and q in a way that leads to the most socially desirable outcomes. The axioms of Section 5.2.2 allow us to give a satisfying answer to this question. Specifically, our main theoretical result is a characterization of $L(p, q)$ aggregation in terms of the three axioms.

Theorem 5.3.1. *$L(p, q)$ aggregation, where $p, q \in [1, \infty]$, satisfies consensus, efficiency, and strategyproofness if and only if $p = q = 1$.*

We remark that for $p = q$, Equation (5.1) does not distinguish between different reviewers, that is, the aggregation method pools all reviews together. We find this interesting, because the $L(p, q)$ aggregation framework does have enough power to make that distinction, but the axioms guide us towards a specific solution, $L(1, 1)$, which does not.

Turning to the proof of the theorem, we start from the easier 'if' direction.

5.3.1 $p = q = 1$ Satisfies All Three Axioms

Lemma 5.3.2. *$L(p, q)$ aggregation with $p = q = 1$ satisfies consensus, efficiency and strategyproofness.*

Proof. The key idea of the proof lies in the form taken by the minimizer of $L(1, 1)$ loss. When each reviewer reviews every paper and the papers have objective criteria scores, $L(1, 1)$ aggregation reduces to computing

$$\widehat{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} \left\{ \sum_{i \in \mathcal{R}} \sum_{a \in \mathcal{P}} |y_{ia} - f(\mathbf{x}_a)| \right\}, \tag{5.4}$$

where ties are broken by picking the minimizer with minimum L_2 norm. We claim that the aggregate function is given by

$$\widehat{f}(\mathbf{x}_a) = \text{left-med}(\{y_{ia}\}_{i \in \mathcal{R}}) \quad \forall a \in \mathcal{P},$$

where left-med$(\cdot)$ of a set of points is their left median. We prove this claim by showing four parts:

(i) $\widehat{f}$ is a valid function,
(ii) $\widehat{f}$ is an unconstrained minimizer of the objective in (5.4),
(iii) $\widehat{f}$ satisfies the constraints of (5.4), i.e., $\widehat{f} \in \mathcal{F}$, and
(iv) $\widehat{f}$ has the minimum L_2 norm among all minimizers of (5.4).

We start by proving part (i). This part can only be violated if there are two papers a and b such that $\mathbf{x}_a = \mathbf{x}_b$, but left-med$(\{y_{ia}\}_{i \in \mathcal{R}}) \neq$ left-med$(\{y_{ib}\}_{i \in \mathcal{R}})$, leading to $\widehat{f}$ having two function values for the same $\mathbf{x}$-value. However, we assumed that each reviewer i has a function $g_i^\star$ used to score the papers. So, for the two papers a and b, we would have $y_{ia} = g_i^\star(\mathbf{x}_a) = g_i^\star(\mathbf{x}_b) = y_{ib}$ for every i, giving us left-med$(\{y_{ia}\}_{i \in \mathcal{R}}) =$ left-med$(\{y_{ib}\}_{i \in \mathcal{R}})$. Therefore, $\widehat{f}$ is a valid function.

For part (ii), consider the optimization problem (5.4) without any constraints. Denote the objective function as $G(f)$. Rearranging terms, we obtain

$$G(f) = \sum_{a \in \mathcal{P}} \sum_{i \in \mathcal{R}} |y_{ia} - f(\mathbf{x}_a)|. \tag{5.5}$$

Consider the inner summation $\sum_{i\in\mathcal{R}} |y_{ia} - f(\mathbf{x}_a)|$; it is obvious that this quantity is minimized when $f(\mathbf{x}_a)$ is any median of the $\{y_{ia}\}_{i\in\mathcal{R}}$ values. Hence, we have

$$\begin{aligned} G(f) &= \sum_{a\in\mathcal{P}} \sum_{i\in\mathcal{R}} |y_{ia} - f(\mathbf{x}_a)| \\ &\geq \sum_{a\in\mathcal{P}} \sum_{i\in\mathcal{R}} |y_{ia} - \text{left-med}(\{y_{ia}\}_{i\in\mathcal{R}})| \\ &= G(\widehat{f}), \end{aligned} \tag{5.6}$$

where f is an arbitrary function. Therefore, $\widehat{f}$ minimizes the objective function even in the absence of any constraints, proving part (ii).

Turning to part (iii), we show that $\widehat{f}$ satisfies the monotonicity constraints, i.e., $\widehat{f} \in \mathcal{F}$. Suppose $a, b \in \mathcal{P}$ are such that $\mathbf{x}_a \geq \mathbf{x}_b$. Using the fact that each reviewer i scores papers based on the function $g_i^\star$, we have $y_{ia} = g_i^\star(\mathbf{x}_a)$ and $y_{ib} = g_i^\star(\mathbf{x}_b)$. And since $g_i^\star \in \mathcal{F}$ obeys monotonicity constraints, we obtain $y_{ia} \geq y_{ib}$ for every i. This trivially implies that left-med$(\{y_{ia}\}_{i\in\mathcal{R}}) \geq$ left-med$(\{y_{ib}\}_{i\in\mathcal{R}})$, i.e., $\widehat{f}(\mathbf{x}_a) \geq \widehat{f}(\mathbf{x}_b)$, completing part (iii).

Finally, we prove part (iv). Observe that Equation (5.6) is a strict inequality if there is a paper a for which $f(\mathbf{x}_a)$ is not a median of the $\{y_{ia}\}_{i\in\mathcal{R}}$ values. In other words, the only functions f that have the same objective function value as $\widehat{f}$ are of the form

$$f(\mathbf{x}_a) \in \text{med}(\{y_{ia}\}_{i\in\mathcal{R}}) \quad \forall a \in \mathcal{P}, \tag{5.7}$$

where med$(\cdot)$ of a collection of points is the set of all points between (and including) the left and right medians. Hence, all other minimizers of (5.4) must satisfy Equation (5.7). Observe that $\widehat{f}$ is pointwise smaller than any of these functions, since it computes the left median at each of the $\mathbf{x}$-values. Therefore, $\widehat{f}$ has the minimum L_2 norm among all possible minimizers of (5.4), completing the proof of part (iv).

Combining all four parts proves that $\widehat{f}$ is indeed the aggregate function chosen by $L(1,1)$ aggregation. We use this to prove that $L(1,1)$ aggregation satisfies consensus, efficiency and strategyproofness.

Consensus. Let $a \in \mathcal{P}$ be a paper such that $y_{1a} = y_{2a} = \cdots = y_{ma} = r$ for some r. Then, left-med$(\{y_{ia}\}_{i\in\mathcal{R}}) = r$. Hence, $\widehat{f}(\mathbf{x}_a) = r$, satisfying consensus.

Efficiency. Let $a, b \in \mathcal{P}$ be such that a dominates b. In other words, the sorted overall recommendations given to a pointwise-dominate the sorted overall recommendations given to b. So, by definition, left-med$(\{y_{ia}\}_{i\in\mathcal{R}})$ is at least as large as left-med$(\{y_{ib}\}_{i\in\mathcal{R}})$. That is, $\widehat{f}(\mathbf{x}_a) \geq \widehat{f}(\mathbf{x}_b)$, satisfying efficiency.

Strategyproofness. Let i be an arbitrary reviewer. Observe that in this setting, the aggregate score $\widehat{f}(\mathbf{x}_a)$ of a paper a depends only on the score y_{ia} and not on other scores $\{y_{ib}\}_{b\neq a}$ given by reviewer i. In other words, the only way to manipulate $\widehat{f}(\mathbf{x}_a) =$ left-med$(\{y_{i'a}\}_{i'\in\mathcal{R}})$ is by changing y_{ia}. Consider three cases. Suppose $y_{ia} < \widehat{f}(\mathbf{x}_a)$. In this case, if reviewer i reports $y'_{ia} \leq \widehat{f}(\mathbf{x}_a)$, then there is no change in the aggregate score of a. On the other hand, if $y'_{ia} > \widehat{f}(\mathbf{x}_a)$, then either the aggregate score of a remains the same or increases, making things only worse for reviewer i. The other case of $y_{ia} > \widehat{f}(\mathbf{x}_a)$ is symmetric to $y_{ia} < \widehat{f}(\mathbf{x}_a)$.

Consider the third case, $y_{ia} = \widehat{f}(\mathbf{x}_a)$. In this case, manipulation can only make things worse since we already have $|y_{ia} - \widehat{f}(\mathbf{x}_a)| = 0$. In summary, reporting y'_{ia} instead of y_{ia} cannot help decrease $|y_{ia} - \widehat{f}(\mathbf{x}_a)|$. Also, recall that y_{ia} does not affect the aggregate scores of other papers, and hence manipulation of y_{ia} does not help them either. Therefore, by manipulating any of the y_{ia} scores, reviewer i cannot bring the aggregate recommendations closer to her own, proving strategyproofness. □

5.3.2 Violation of the Axioms When $(p, q) \neq (1, 1)$

We now tackle the harder 'only if' direction of Theorem 5.3.1. We do so in three steps: efficiency is violated by $p \in (1, \infty)$ and $q = 1$ (Lemma 5.3.3), strategyproofness is violated by $L(p, q)$ aggregation for all $q > 1$ (Lemma 5.3.4), and consensus is violated by $p = \infty$ and $q = 1$ (Lemma 5.3.5). Together, the three lemmas leave $p = q = 1$ as the only option. Below we state the lemmas and give some proof ideas; the theorem's full proof is relegated to Appendix D.1.

It is worth noting that, although we have presented the lemmas as components in the proof of Theorem 5.3.1, they also have standalone value (some more than others). For example, if one decided that only strategyproofness is important, then Lemma 5.3.4 below would give significant guidance on choosing an appropriate method.

Violation of efficiency

In our view, the following lemma presents the most interesting and counter-intuitive result in the paper.

Lemma 5.3.3. *$L(p, q)$ aggregation with $p \in (1, \infty)$ and $q = 1$ violates efficiency.*

It is quite surprising that such reasonable loss functions violate the simple requirement of efficiency. In what follows we attempt to explain this phenomenon via a connection between our problem and the notion of the 'Fermat point' of a triangle [Spa96]. The explanation provided here demonstrates the negative result for $L(2, 1)$ aggregation. The complete proof of the lemma for general values of $p \in (1, \infty)$ is quite involved, as can be seen in Appendix D.1.

Consider a setting with 3 reviewers and 2 papers, where each reviewer reviews both papers. We let $\mathbf{x}_1$ and $\mathbf{x}_2$ denote the respective objective criteria scores of the two papers. Assume that no score in $\{\mathbf{x}_1, \mathbf{x}_2\}$ is pointwise greater than or equal to the other score in that set. Let the overall recommendations given by the reviewers be $y_{11} = z$, $y_{21} = 0$, $y_{31} = 0$ to the first paper and $y_{12} = 0$, $y_{22} = 1$ and $y_{23} = 0$ to the second paper. Under these scores, let $\widehat{f}$ denote the aggregate function that minimizes the $L(2, 1)$ loss.

The Fermat point of a triangle is a point such that the sum of its (Euclidean) distances from all three vertices is minimized. Consider a triangle in $\mathbb{R}^2$ with vertices $(z, 0)$, $(0, 1)$ and $(0, 0)$. Setting $z = 2$, one can use known algorithms to compute the Fermat point of this triangle as $(0.25, 0.30)$. More generally, when the vertex $(z, 0)$ is moved away from the rest of the triangle (by increasing z), the Fermat point paradoxically biases towards the other (second) coordinate.

Connecting back to our original problem, by definition, the Fermat point of this triangle is exactly $(\widehat{f}(\mathbf{x}_1), \widehat{f}(\mathbf{x}_2))$. When $z = 2$, paper 1 receives scores $(2, 0, 0)$ in sorted order, which dominates the sorted scores $(1, 0, 0)$ of paper 2. However the aggregate score $\widehat{f}(\mathbf{x}_1) = 0.25$ of paper 1 is strictly smaller than $\widehat{f}(\mathbf{x}_2) = 0.30$ of paper 2, thereby violating efficiency for the $L(2, 1)$ loss.

As a final but important remark, the proof of Lemma 5.3.3 only requires a significantly weaker notion of efficiency. In this weaker notion, we consider two papers a and b such that their reviews are symmetric (formally, switching the labels a and b and switching the labels of some reviewers leaves the data unchanged). In this case, reducing the review scores of paper b must lead to $\widehat{f}(\mathbf{x}_a) \geq \widehat{f}(\mathbf{x}_b)$.

Violation of strategyproofness

Lemma 5.3.4. *$L(p, q)$ aggregation with $q \in (1, \infty]$ violates strategyproofness.*

We prove the lemma via a simple construction with just one paper and two reviewers, who give the paper overall recommendations of 1 and 0, respectively. For $q \in (1, \infty)$, the aggregate score is

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathbb{R}} \Big\{ |1 - f|^q + |f|^q \Big\},$$

and for $q = \infty$, it is

$$\widehat{f} = \operatorname*{argmin}_{f \in \mathbb{R}} \max\big(|1 - f|, |f|\big).$$

Either way, the unique minimum is obtained at an aggregate score of 0.5. If reviewer 1 reported an overall recommendation of 2, however, the aggregate score would be 1, which matches her 'true' recommendation, thereby violating strategyproofness. See Appendix D.1.2 for the complete proof.

Violation of consensus

Lemma 5.3.5. *$L(p, q)$ aggregation with $p = \infty$ and $q = 1$ violates consensus.*

Lemma 5.3.5 is established via another simple construction: two papers, two reviewers, and overall recommendations

$$\mathbf{y} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix},$$

where y_{ia} denotes the overall recommendation given by reviewer i to paper a. Crucially, the two reviewers agree on an overall recommendation of 1 for paper 2, hence the aggregate score of this paper must also be 1. But we show that $L(\infty, 1)$ aggregation would *not* return an aggregate score of 1 for paper 2. The formal proof appears in Appendix D.1.3.

5.4 Implementation and Experimental Results

In this section, we provide an empirical analysis of a few aspects of peer review through the approach of this paper. We employ a dataset of reviews from the 26$^{\text{th}}$ International

# of reviews by a reviewer	1	2	3	4	5	6	7	8	≥ 9
Frequency	238	96	92	120	146	211	628	187	7

Table 5.1: Distribution of number of papers reviewed by a reviewer.

Joint Conference on Artificial Intelligence (IJCAI 2017), which was made available to us by the program chair. To our knowledge, we are the first to use this dataset.

At submission time, authors were asked if review data for their paper could be included in an anonymized dataset, and, similarly, reviewers were asked whether their reviews could be included; the dataset provided to us consists of all reviews for which permission was given. Each review is tagged with a reviewer ID and paper ID, which are anonymized for privacy reasons. The criteria used in the conference are 'originality', 'relevance', 'significance', 'quality of writing' (which we call 'writing'), and 'technical quality' (which we call 'technical'), and each is rated on a scale from 1 to 10. Overall recommendations are also on a scale from 1 to 10. In addition, information about which papers were accepted and which were rejected is included in the dataset.

The number of papers in the dataset is 2380, of which 649 were accepted, which amounts to 27.27%. This is a large subset of the 2540 submissions to the conference, of which 660 were accepted, for an actual acceptance rate of 25.98%. The number of reviewers in the dataset is 1725, and the number of reviews is 9197. All but nine papers in the dataset have three reviews (485 papers), four reviews (1734 papers), or five reviews (152) papers. Table 5.1 shows the distribution of the number of papers reviewed by reviewers.

We apply $L(1,1)$ aggregation (i.e., $p = q = 1$), as given in Equation (5.1), to this dataset to learn the aggregate function. Let us denote that function by $\widetilde{f}$. The optimization problem in Equation (5.1) is convex, and standard optimization packages can efficiently compute the minimizer. Hence, importantly, computational complexity is a nonissue in terms of implementing our approach.

Once we compute the aggregate function $\widetilde{f}$, we calculate the aggregate overall recommendation of each paper a by taking the median of the aggregate reviewer scores for that paper obtained by applying $\widetilde{f}$ to the objective scores:

$$y_{\widetilde{f}}(a) = \text{median}(\{\widetilde{f}(\mathbf{x}_{ia})\}_{i \in R(a)}) \quad \forall a \in \mathcal{P}. \tag{5.8}$$

Recalling that 27.27% of the papers in the dataset were actually accepted to the conference, in our experiments we define the set of papers accepted by the aggregate function $\widetilde{f}$ as the the top 27.27% of papers according to their respective $y_{\widetilde{f}}$ values. We now present the specific experiments we ran, and their results.

5.4.1 Varying Number of Reviewers

In our first experiment, for each value of a parameter $k \in \{1, \ldots, 5\}$, we subsampled k distinct reviews for each paper uniformly at random from the set of all reviews for that paper (if the paper had fewer than k to begin with then we retained all the reviews). We then computed an aggregate function, $\widehat{f}_k$, via $L(1,1)$ aggregation applied only to these

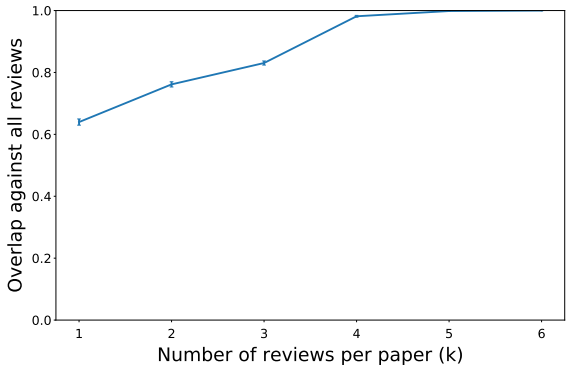


Figure 5.1: Fraction overlap as number of reviews per paper is restricted. Error bars depict 95% confidence intervals, but may be too small to be visible for $k = 4, 5$.

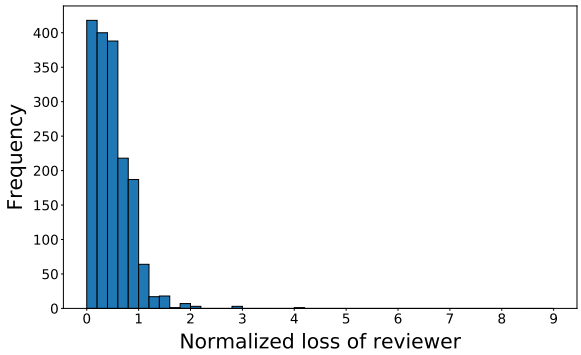


Figure 5.2: Frequency of losses of the reviewers for $L(1,1)$ aggregation, normalized by the number of papers reviewed by the respective reviewers.

subsampled reviews. Next, we found the set of top 27.27% papers as given by $\widehat{f}_k$ applied to the subsampled reviews. Finally, we compared the overlap of this set of top papers for every value of k with the set of top 27.27% papers as dictated by the overall aggregate function $\widetilde{f}$.

The results from this experiment are plotted in Figure 5.1, and lead to several observations. First, the incremental overlap from $k = 4$ to 5 is very small because there are very few papers that had 5 or more reviews. Second, we see that the amount of overlap monotonically increases with the number of reviewers per paper k, thereby serving as a sanity check on the data as well as our methods. Third, we observe the overlap to be quite high ($\approx 60\%$) even with a single reviewer per paper.

5.4.2 Loss Per Reviewer

Next, we look at the loss of different reviewers, under $\widetilde{f}$ (obtained by $L(1,1)$ aggregation). In order for the losses to be on the same scale, we normalize each reviewer's loss by the number of papers reviewed by them. Formally, the normalized loss of reviewer i (for $p = 1$) is

$$\frac{1}{|P(i)|} \sum_{a \in P(i)} |y_{ia} - \widetilde{f}(\mathbf{x}_{ia})|.$$

The normalized loss averaged across reviewers is found to be 0.470, and the standard deviation is 0.382. Figure 5.2 shows the distribution of the normalized loss of all the reviewers. Note that the normalized loss of a reviewer can fall in the range $[0, 9]$. These results thus indicate that the function $\widetilde{f}$ is indeed at least a reasonable representation of the mapping of the broader community.

5.4.3 Overlap of Accepted Papers

We also compute the overlap between the set of top 27.27% papers selected by $L(1,1)$ aggregation $\widetilde{f}$ with the *actual* 27.27% accepted papers. It is important to emphasize that we believe the set of papers selected by our method is *better* than any hand-crafted or rule-based decision using the scores, since this aggregate represents the opinion of the community. Hence, to be clear, we do *not* have a goal of maximizing the overlap. Nevertheless, a very small overlap would mean that our approach is drastically different from standard practice, which would potentially be disturbing. We find that the overlap is 79.2%, which we think is quite fascinating — our approach does make a significant difference, but the difference is not so drastic as to be disconcerting.

Out of intellectual curiosity, we also computed the pairwise overlaps of the papers accepted by $L(p,q)$ aggregation, for $p, q \in \{1,2,3\}$. We find that the choice of the reviewer-norm hyperparameter q has more influence than the paper-norm hyperparameter p; we refer the reader to Appendix D.2.1 for details. Finally, in Appendix D.2.2 we present visualizations of $L(1,1)$ aggregation, which provide insights into the preferences of the community.

5.5 Discussion

We address the problem of subjectivity in peer review by combining approaches from machine learning and social choice theory. A key challenge in the setting of peer review (e.g., when choosing a loss function) is the absence of ground truth, and we overcome this challenge via a principled, axiomatic approach.

One can think of the theoretical results of Section 5.3 as supporting $L(1,1)$ aggregation using the tools of social choice theory, whereas the empirical results of Section 5.4 focus on studying its *behavior* on real data. Understanding this helps clear up another possible source of confusion: are we not overfitting by training on a set of reviews, and then applying the aggregate function to the same reviews? The answer is negative, because the process of learning the function $\widehat{f}$ amounts to an aggregation of opinions about how criteria scores should be mapped to overall recommendations. Applying it to the data yields recommendations in $\mathbb{Y}$, whereas this function from $\mathbb{X}$ to $\mathbb{Y}$ lives in a different space.

That said, it is of intellectual interest to understand the statistical aspects of estimating the community's consensus mapping function, assuming the existence of a ground truth. In more detail, suppose that each reviewer's true function $g_i^\star$ is a noisy version of some underlying function $f^{\star\star}$ that represents the community's beliefs. Then can $L(1,1)$ aggregation recover the function $f^{\star\star}$ (in the statistical consistency sense)? If so, then with what sample complexity? At a conceptual level, this non-parametric estimation problem is closely related to problems in isotonic regression [Sha+16; GW07; CGS18]. The key difference is that the observations in our setting consist of evaluations of multiple functions, where each such function is a noisy version of the original monotonic function. In contrast, isotonic regression is primarily concerned with noisy evaluations of a common function. Nevertheless, the insights from isotonic regression suggest that the naturally occurring monotonicity

assumption of our setting can yield attractive — and sometimes near-parametric [Sha+16; SBW18] — rates of estimation.

Our work focuses on learning one representative aggregate mapping for the entire community of reviewers. Instead, the program chairs of a conference may wish to allow for multiple mappings that represent the aggregate opinions of different sub-communities (e.g., theoretical or applied researchers). In this case, one can modify our framework to also learn this (unknown) partition of reviewers and/or papers into multiple sub-communities with different mapping functions, and frame the problem in terms of learning a mixture model. The design of computationally efficient algorithms for $L(p, q)$ aggregation under such a mixture model is a challenging open problem.

As a final remark, we see our work as an unusual synthesis between computational social choice and machine learning. We hope that our approach will inspire exploration of additional connections between these two fields of research, especially in terms of viewing choices made in machine learning — often in an *ad hoc* fashion — through the lens of computational social choice.

Chapter 6

Axioms for Learning from Pairwise Comparisons

To be well-behaved, systems that process preference data must satisfy certain conditions identified by economic decision theory and by social choice theory. In ML, preferences and rankings are commonly learned by fitting a probabilistic model to noisy preference data. The behavior of this learning process from the view of economic theory has previously been studied for the case where the data consists of rankings. In practice, it is more common to have only pairwise comparison data, and the formal properties of the associated learning problem are more challenging to analyze. We show that a large class of random utility models (including the Thurstone–Mosteller Model), when estimated using the MLE, satisfy a Pareto efficiency condition. These models also satisfy a strong monotonicity property, which implies that the learning process is responsive to input data. On the other hand, we show that these models fail certain other consistency conditions from social choice theory, and in particular do not always follow the majority opinion. Our results inform existing and future applications of random utility models for societal decision making.

6.1 Introduction

More than two centuries ago, the marquis de Condorcet [Con85] suggested a statistical interpretation of voting. Each vote, Condorcet argued, can be seen as a noisy estimate of a ground-truth ranking of the alternatives. A voting rule should aggregate votes into a ranking that is most likely to coincide with the ground truth. Although Condorcet put forward a specific noise model, his reasoning applies to any *random noise model*, which is a distribution over votes parameterized by a ground truth ranking [CS05; CPS16].

Until NeurIPS 2014, this statistical approach to voting was studied in parallel to the more common normative approach, which evaluates voting rules based on axiomatic properties. But the two approaches converged in a paper by Azari Soufiani et al. [APX14a], whose key idea was to determine whether maximum likelihood estimators (MLEs) for two noise models satisfy basic axiomatic properties. Their results were sharpened and extended

by Xia [Xia16].

Our point of departure is that instead of random noise models we consider *random utility models*, where each alternative x has a utility β_x, and the probability of drawing a pairwise comparison that puts x above y depends on β_x and β_y. For example, under the well-known Thurstone–Mosteller Model [Thu27; Mos51], this pairwise comparison would be generated by sampling $u(x)$ and $u(y)$ from normal distributions with means β_x and β_y, respectively, and the same variance.

Our research question, then, is this:

> *For a given random utility model, consider the aggregation rule that takes pairwise comparisons between alternatives as input and returns the ranking over alternatives defined by the MLE; which axioms does it satisfy?*

6.1.1 Why Is the Research Question Important?

The MLE, as an aggregation rule, is *statistically* well-motivated. From a voter's perspective, though, it is not immediately clear the the MLE is a good rule that adequately aggregates preferences. In particular, in case the statistical assumptions of a random utility fail to capture reality, the MLE may give a bad result. However, if we can show that the MLE satisfies standard axioms from voting theory, this implies a certain degree of robustness. It also provides reassurance that the statistical process cannot fall prey to pathological behavior in edge cases.

A string of recent papers [Fre+20; Noo+18; Kah+19; Lee+19] proposes a sequence of systems for automated societal decision making through social choice and machine learning. They all aggregate pairwise comparisons, by fitting them to random utility models. As these systems are being deployed to support decisions in non-profits and government, it becomes crucial to understand normative properties of this framework.[1]

The work of Freedman et al. [Fre+20] provides a concrete illustration. The paper deals with prioritization of patients in kidney exchange. They asked workers on Amazon Mechanical Turk to decide which of each given pair of patients with chronic kidney disease (defined by their medical profiles) should receive a kidney, and computed the MLE utilities assuming the pairwise comparisons were generated by a Bradley–Terry model [Bra84]. The resulting ranking over profiles was used to help a kidney exchange algorithm prioritize some patients over others. As is common, the authors pooled pairwise comparisons reported by many different voters, and fit a single random utility model to the pool, as opposed to fitting models to individual voters. This move is usually done to improve statistical accuracy, but, to an extent, it invalidates the underlying motivation of the noise model, which imagines a single decision maker with imprecise perception of utilities. Pooling assumes that a group of agents can be captured by the same model. Given this leap of faith, we believe that a normative analysis of the process becomes especially important.

Other papers apply an emerging approach called *virtual democracy* [Kah+19] to automate decisions in two domains, autonomous vehicles [Noo+18] and food allocation [Lee+19].

[1] Previous work [APX14a; Xia16] does not apply as it focuses on the aggregation of input *rankings* (a special case of pairwise comparisons) through random noise models.

Lee et al. [Lee+19] asked stakeholders in a nonprofit food rescue organization to report which of each given pair of recipient organizations should be allocated an incoming food donation. Unlike Freedman et al. [Fre+20], they fit a random utility model (Thurstone–Mosteller) to the pairwise comparisons provided by each stakeholder *individually*, and used the Borda voting rule to aggregate the predictions given by each of the individual models. On the one hand, our axiomatic results may justify a move to the pooled approach of Freedman et al. [Fre+20], which could improve accuracy. On the other hand, even when learning individual models, axiomatics can convince voters that their preferences are learned using a sensible method.

6.1.2 Our Results

We examine four axiomatic properties, suitably adapted to our setting. Informally, they are:

- *Pareto efficiency:* If x dominates y in the input dataset, x should be above y in the MLE ranking.
- *Monotonicity:* Adding $a \succ b$ comparisons to the input dataset can only help a and harm b in the MLE ranking.
- *Pairwise majority consistency:* If the input dataset is consistent with a ranking over the alternatives, that ranking must coincide with the MLE ranking.
- *Separability:* If a is preferred to b in the MLE rankings of two different datasets, a must also be preferred to b in the MLE ranking of the combined dataset.

The first two properties, Pareto efficiency and monotonicity, have immediate appeal and seem crucial: a system violating these is not faithful to input preferences. In Sections 6.3 and 6.4 we show that both properties are satisfied by a large class of random utility models when fitted using MLEs. For monotonicity, our main result, the proof is surprisingly involved, since we need to reason about the optimum utility values of all alternatives simultaneously. (In contrast, for random noise models, monotonicity is a simple consequence of the definition [APX14a; Xia16].)

The latter two properties are *not* satisfied by MLEs, for all random utility models satisfying mild conditions. In a way, these negative results illuminate the behavior of random utility models: The case of pairwise majority consistency illustrates a trade-off, where random utility models ensure that a strong preference is respected, even if this leads them to override a majority preference elsewhere. While negative, we do not see the counterexamples in Sections 6.5 and 6.6 as pathological, though they may suggest contexts in which the use of random utility models is not appropriate.

6.2 Model

Let $\mathcal{X}$ be a finite set of alternatives. For notational simplicity, we let $\mathcal{X}^2 = \{(x, y) : x, y \in \mathcal{X}, x \neq y\}$ denote the set of *distinct* pairs of alternatives. Let $\# : \mathcal{X}^2 \to \mathbb{N}$ be a *dataset* of

pairwise comparisons between alternatives: For $x, y \in \mathcal{X}$, $\#\{x \succ y\}$ is the number of times x beat y in the dataset.

The (pairwise) *comparison graph* $\mathcal{G}_\# = (\mathcal{X}, E)$ with respect to dataset $\#$ is the directed graph with the alternatives $\mathcal{X}$ as the vertices, and edges E such that there exists a directed edge $(u, v) \in E$ iff $\#\{u \succ v\} > 0$. We say that $\mathcal{G}_\#$ is *connected* if its undirected form is connected, and we call it *strongly connected* if for all $(x, y) \in \mathcal{X}^2$, there is a directed path from x to y in $\mathcal{G}_\#$.

Given a dataset, our goal is to learn a random utility model (RUM). A random utility model specifies, for any two distinct alternatives $x, y \in \mathcal{X}$, the probability that when asking the decision maker to compare x and y, the answer will be $x > y$. (Due to noise, when repeatedly querying the same pair, we may see different answers.) For us, a random utility model is parameterized by a vector $\beta \in \mathbb{R}^{\mathcal{X}}$, where β_x is an unknown utility value for $x \in \mathcal{X}$. When we ask for a comparison between two alternatives $x, y \in \mathcal{X}$, we model the decision maker as sampling noisy utilities $u(x)$ and $u(y)$ from distributions parameterized by (and typically centered at) β_x and β_y. Then, the decision maker reports the comparison $x > y$ iff $u(x) > u(y)$.

In this paper, we focus on random utility models with i.i.d. noise, so that $u(x) = \beta_x + \zeta(x)$, where $\zeta(x) \sim \mathcal{P}$ is i.i.d. across all alternatives. Let F be the CDF of a random variable which is the difference between two independent random variables with distribution $\mathcal{P}$. Then the probability that alternative x beats y when they are compared is[2]

$$\Pr(x \succ y) = \Pr(u(x) > u(y)) = \Pr(\zeta(y) - \zeta(x) < \beta_x - \beta_y) = F(\beta_x - \beta_y). \tag{6.1}$$

We derived Equation (6.1) from a specific noise model, but it makes sense for any function $F : \mathbb{R} \to [0, 1]$ with CDF-like properties, even if it does not correspond to a noise distribution $\mathcal{P}$. Indeed, we can take any F which is non-decreasing, satisfies $F(\Delta u) + F(-\Delta u) = 1$ for all $\Delta u \in \mathbb{R}$, and is such that $\lim_{\Delta u \to -\infty} F(\Delta u) = 0$ and $\lim_{\Delta u \to \infty} F(\Delta u) = 1$. We adopt Equation (6.1) as the general definition of a random utility model for our technical results.

Two of the most common random utility models are

- the *Thurstone–Mosteller (TM) model*: We sample utility as $u(x) = \beta_x + \zeta(x)$, with i.i.d. noise $\zeta(x) \sim \mathcal{N}(0, 1/2)$. This is equivalent to Equation (6.1) with F as the Gaussian CDF Φ.
- the *Bradley–Terry model* (equivalent to the *Plackett–Luce* model restricted to pairwise comparisons), where $\Pr(x \succ y) = \frac{e^{u(x)}}{e^{u(x)} + e^{u(y)}}$. This is Equation (6.1) with F as the logistic function.

We usually assume that F is strictly log-concave, and that it is strictly monotonic and continuous,[3] so that F has an inverse on $(0, 1)$. These conditions hold for Thurstone–Mosteller and Bradley–Terry.

For a random utility model, given a dataset $\#$, our goal is to find parameters $(\beta_x)_{x \in \mathcal{X}}$ that best fit $\#$. We find these parameters by maximum likelihood estimation. The log-

[2] We assume $\mathcal{P}$ to be a continuous distribution, and so we do not have to worry about ties.

[3] Continuity of F is guaranteed when the corresponding noise distribution $\mathcal{P}$ is continuous.

likelihood is given by

$$\mathcal{L}(\beta) = \sum_{(x,y)\in\mathcal{X}^2} \#\{x \succ y\} \log F(\beta_x - \beta_y).$$

When the dataset $\#$ is clear from the context, we write $\hat{\beta} \in \mathbb{R}^{\mathcal{X}}$ for a parameter vector that maximizes log-likelihood, and say that $\hat{\beta}$ is the *MLE*. Note that if $c \in \mathbb{R}$ is a scalar, then $\mathcal{L}(\beta) = \mathcal{L}(\beta + c)$ for all $\beta \in \mathbb{R}^{\mathcal{X}}$ (since $\Pr(x \succ y)$ depends only on the difference $\beta_x - \beta_y$), so the MLE is only defined up to an additive shift. For concreteness, we pick some $r \in \mathcal{X}$, call it the *reference alternative*, and fix $\beta_r = 0$; then, we maximize $\mathcal{L}$ over $\mathcal{D} = \{\beta \in \mathbb{R}^{\mathcal{X}} : \beta_r = 0\}$.

A random utility model is particularly appropriate when the dataset $\#$ consists of pairwise comparisons which are all reported by a single decision maker. However, in many cases the dataset is obtained by pooling reports from many agents, for instance to minimize the labeling effort of each individual agent, or if we have the explicit aim to aggregate preferences from different agents. Some of the axioms we study are explicitly motivated by cases where $\# = \sum_{i\in\mathcal{R}} \#_i$, i.e., the dataset is obtained by pooling individual datasets, where $\mathcal{R}$ is the set of agents. It then seems natural to assume that each agent behaves in accordance with some random utility model with unknown parameters β^i and unknown CDF-like function F_i. Then the dataset $\#_i$ is generated by repeatedly querying the agent's random utility model for a comparison.

6.2.1 Existence and Boundedness of MLE

Before turning to our main results, we briefly state conditions that guarantee the existence of a finite MLE, and that guarantee uniqueness (up to a shift).

In some scenarios, no finite β maximizes likelihood, and thus the MLE may not exist. For instance, if some alternative a beats other alternatives, but is not beaten even a single time in the dataset, the likelihood can always be strictly increased by increasing β_a (when F is strictly monotonic). Lemma 6.2.1 states a condition under which an MLE exists (i.e. $\mathcal{L}(\beta)$ has a maximizer). Its proof also provides a weak bound on one such maximizer. The proofs of the results in this section are in Appendix E.1.

Lemma 6.2.1 (MLE exists)**.** *Suppose F is strictly monotonic and continuous. Then the MLE exists if and only if every connected component of the comparison graph $\mathcal{G}_{\#}$ is strongly connected.*

For alternative $x, y \in \mathcal{X}$, we define the *perfect-fit distance* between x and y as

$$\delta(x, y) := F^{-1}\left(\frac{\#\{x \succ y\}}{\#\{x \succ y\} + \#\{y \succ x\}}\right).$$

This is the difference in utilities of x and y required for the model to exactly match the observed frequencies of $\#\{x \succ y\}$ and $\#\{y \succ x\}$ in the data. We can check that the MLE will respect this perfect-fit distance when an alternative has only a single neighbor in the comparison graph.

Lemma 6.2.2. *Let F be strictly monotonic and continuous. Suppose that for alternative a there is exactly one alternative b for which $\#\{a \succ b\}+\#\{b \succ a\} > 0$. If both $\#\{a \succ b\} > 0$ and $\#\{b \succ a\} > 0$, then any MLE $\hat{\beta}$ satisfies $\hat{\beta}_a - \hat{\beta}_b = \delta(a, b)$.*

We can use this result to provide a stronger bound on the MLE than the one from Lemma 6.2.1, which holds under slightly stronger conditions.

Lemma 6.2.3. *Suppose that $\#\{x \succ y\} > 0$ and $\#\{y \succ x\} > 0$ for all x and y, and that F is continuous and strictly monotonic. Then for every MLE $\hat{\beta}$ we have the bound*

$$\|\hat{\beta}\|_\infty \leq |\mathcal{X}| \cdot \max_{(x,y)\in\mathcal{X}^2} \delta(x, y).$$

6.2.2 Uniqueness of MLE

Under mild conditions on the function F and the comparison graph $\mathcal{G}_\#$, we have seen that a bounded MLE exists. When is the MLE unique? Note that if F is a strictly log-concave, this implies that the log-likelihood $\mathcal{L}(\beta)$ is concave. If we additionally require that the comparison graph $\mathcal{G}_\#$ is connected, then $\mathcal{L}(\beta)$ is in fact strictly concave, and thus the MLE is unique, as we prove in Appendix E.2.

Lemma 6.2.4. *Suppose that F is strictly log-concave. Then $\mathcal{L}(\beta)$ is strictly concave and the MLE is unique (assuming it exists), if and only if the comparison graph $\mathcal{G}_\#$ is connected.*

6.3 Pareto Efficiency

A minimal requirement in economic theory is *Pareto efficiency*: if all agents prefer a to b, then in aggregate, a should be preferred to b. A first attempt at defining this notion for the environment of pairwise comparisons would be to say that if $\#\{a \succ b\} > 0$ but $\#\{b \succ a\} = 0$, then the MLE should satisfy $\hat{\beta}_a \geq \hat{\beta}_b$. However, this property is too restrictive. Consider a dataset with

$$\#\{a \succ b\} = 100, \#\{b \succ c\} = 1, \#\{c \succ a\} = 1,$$

and all other comparisons 0. To satisfy the mentioned property, the MLE would need to satisfy $\hat{\beta}_a \geq \hat{\beta}_b \geq \hat{\beta}_c \geq \hat{\beta}_a$, so they are all equal; however it seems better to have $\hat{\beta}_a > \hat{\beta}_b$.

A more sensible version of Pareto efficiency is motivated by the multi-agent setting described in Section 6.2, where $\# = \sum_{i\in\mathcal{R}} \#_i$, and each individual dataset $\#_i$ is generated by a random utility model with unknown parameters β^i. In this case, Pareto efficiency should say that if $\beta^i_a > \beta^i_b$ for all $i \in \mathcal{R}$, then the MLE $\hat{\beta}$ applied to dataset $\#$ should satisfy $\hat{\beta}_a > \hat{\beta}_b$ as well. Our official definition of Pareto efficiency implies this, but is phrased more generally.

Definition 6.3.1 (Pareto efficiency)**.** Suppose $a, b \in \mathcal{X}$ satisfy $\#\{a \succ b\} > \#\{b \succ a\}$, and are such that for every other alternative $x \in \mathcal{X} \setminus \{a, b\}$, we have

$$\#\{a \succ x\} > \#\{b \succ x\} \quad \text{and} \quad \#\{x \succ a\} < \#\{x \succ b\}.$$

Then, Pareto efficiency requires that $\hat{\beta}_a \geq \hat{\beta}_b$.

To see that this definition captures the desired behavior in the multi-agent case, note that if $\beta_a^i > \beta_b^i$, then the dataset $\#_i$ satisfies the condition of Definition 6.3.1 with high probability as we grow the number of comparisons in $\#_i$, and similarly the condition holds for the pooled dataset $\# = \sum_{i \in \mathcal{R}} \#_i$.

This version of Pareto efficiency is feasible; in fact, it is satisfied by most random utility models.

Theorem 6.3.2. *Maximum likelihood estimation satisfies Pareto efficiency if F is strictly monotonic.*

The key idea behind the proof (given in Appendix E.3) is that if $a, b \in \mathcal{X}$ satisfy the condition of Definition 6.3.1 but some MLE $\hat{\beta}$ puts $\hat{\beta}_a < \hat{\beta}_b$, then the parameter vector β equal to $\hat{\beta}$ except that $\beta_a = \hat{\beta}_b$ and $\beta_b = \hat{\beta}_a$ has strictly higher log-likelihood.

6.4 Monotonicity

If we add a pairwise comparison $a \succ b$ to a dataset, we should deduce that a is stronger and b is weaker relative to our previous estimates. It would be paradoxical if, upon seeing evidence that a is strong and b is weak, we decided to lower a's utility or increase b's utility. Monotonicity requires that this can never happen. We consider a strong form of this axiom, which requires that a is strengthened relative to *every* other alternative, and not just relative to b.

Definition 6.4.1 (Monotonicity)**.** Suppose that $\#$ and $\tilde{\#}$ are two datasets with unique MLEs $\hat{\beta}$ and $\tilde{\beta}$. Suppose that $\tilde{\#}\{x \succ y\} = \#\{x \succ y\}$ for all $x, y \in \mathcal{X}$ except that $\tilde{\#}\{a \succ b\} > \#\{a \succ b\}$. Then, monotonicity requires that for all $x \in \mathcal{X}$,

$$\tilde{\beta}_a - \tilde{\beta}_x \geq \hat{\beta}_a - \hat{\beta}_x \quad \text{and} \quad \tilde{\beta}_b - \tilde{\beta}_x \leq \hat{\beta}_b - \hat{\beta}_x.$$

Equivalently, monotonicity requires that if $\#\{a \succ b\}$ *decreases*, then a becomes weaker relative to other alternatives, and b becomes stronger. We can interpret monotonicity as guaranteeing a kind of *participation incentive*: If we ask an agent to compare a to b, the agent is assured that the answer can only influence our inferred utilities in the desired direction.

Monotonicity is foundational to the idea of aggregating pairwise comparisons; in a sense, it encodes the proper meaning of a comparison "$a \succ b$". It may be surprising, then, that it is difficult to prove that MLEs of random utility models satisfy monotonicity.[4] While it is easy to check that the difference $\hat{\beta}_a - \hat{\beta}_b$ is increasing in $\#\{a \succ b\}$, it is much trickier to analyze the behavior of the log-likelihood for the positioning of alternatives other than a and b. However, it turns out that random utility models do satisfy the strong monotonicity axiom. Our proof depends crucially on the assumption that F is log-concave. Due to the conceptual importance of monotonicity, we consider this our main result.

Theorem 6.4.2. *Maximum likelihood estimation satisfies monotonicity if F is strictly monotonic, log-concave, and differentiable.*

[4]For the Bradley–Terry model, monotonicity is easier to check, since the first-order conditions of likelihood maximization in that model are well-behaved [GHL14, Prop. 6.3]; that proof does not generalize to other models.

The proof, given in Appendix E.4, is relatively unwieldy. For intuition, let us provide an outline of a proof for the special case of three alternatives a, b, c. Let $\#$ and $\tilde{\#}$ be datasets that are identical except that $\#\{a \succ b\} < \tilde{\#}\{a \succ b\}$, and let $\hat{\beta}$ and $\tilde{\beta}$ be the respective MLEs, which are unique by Lemma 6.2.4. We take a as reference, so $\hat{\beta}_a = \tilde{\beta}_a = 0$. It is easy to see that $\hat{\beta}_b \geq \tilde{\beta}_b$, since otherwise $\hat{\beta}$ would have greater log-likelihood than $\tilde{\beta}$ for the dataset $\tilde{\#}$, as $\hat{\beta}$ performs better on the a vs b comparisons, and performs no worse on other comparisons by optimality for $\#$. To see that also $\hat{\beta}_c \geq \tilde{\beta}_c$, consider first the dataset $\#$ and associated log-likelihood $\mathcal{L}(\beta_b, \beta_c)$ (with β_a fixed to 0). Now, for $x \in \mathbb{R}$, let $\psi(x)$ denote the value of β_c that maximizes $\mathcal{L}(x, \beta_c)$, i.e., maximizes likelihood among parameters β with $\beta_a = 0$ and $\beta_b = x$. One can show that, since F is strictly log-concave, $\psi(x)$ is increasing in x.[5] Notice that the number of comparisons between a and b in a dataset does not influence the optimum position of β_c, once β_a and β_b are fixed. Hence, the function ψ is the same whether defined for $\#$ or for $\tilde{\#}$, since they only differ in a vs b comparisons. We have already seen that $\tilde{\beta}_b \leq \hat{\beta}_b$. Since ψ is increasing, we have $\tilde{\beta}_c = \psi(\tilde{\beta}_b) \leq \psi(\hat{\beta}_b) = \hat{\beta}_c$, proving monotonicity.

To visualize monotonicity, consider the three examples in Figure 6.1. For four alternatives, we generated random datasets by choosing $\#\{x \succ y\}$ uniformly at random between 1 and 100, and picked three examples. In each case, we let $\#\{b \succ c\}$ vary from 0 to 100 (going horizontally from left to right), and show how the MLE of the Thurstone–Mosteller model changes as the number of $b \succ c$ comparisons grows; we fix $\hat{\beta}_a = 0$ as reference. As predicted by Theorem 6.4.2, the orange line of $\hat{\beta}_b$ is increasing, while the green line of $\hat{\beta}_c$ is decreasing. Note that the change in $\#\{b \succ c\}$ can affect other alternatives; in the middle figure, the relative positions of a and d swap.

In the pooled setting $\# = \sum_{i \in \mathcal{R}} \#_i$ of Section 6.2, where each agent $i \in \mathcal{R}$ is described by a random utility model with parameters β^i that generates $\#_i$, a natural notion of monotonicity is this: Suppose we calculate the MLE $\hat{\beta}$ for $\#$ and suppose we increase the utility β^i_a for some agent i and some alternative a while keeping all other parameters fixed. Then the updated MLE $\tilde{\beta}$ should satisfy $\tilde{\beta}_a - \tilde{\beta}_x \geq \hat{\beta}_a - \hat{\beta}_x$ for all $x \in \mathcal{X}$: the learned utility of a increases relative to other alternatives. Theorem 6.4.2 implies that random utility models (subject to the theorem's conditions) satisfy this pooled monotonicity notion with high probability, when $\#_i$ consists of many samples and when the number of comparisons is uniform across pairs. The reason is this: with high probability, the increase of β^i_a increases the number of $a \succ x$ comparisons in $\#_i$ for all x. Assuming for now that no other dataset $\#_j$ and no other pairs in $\#_i$ are affected, then successively invoking Theorem 6.4.2 on $a \succ x$ pairs yields the result. Now, with some probability, other parts will be affected, but not too much. Since the MLE is continuous in $\#$ (see Appendix E.5), this noise will not invalidate monotonicity.

[5] Assume that F is twice differentiable. Since $\log F$ is strictly concave, its second derivative is strictly negative. A straightforward calculation shows that then $\partial^2\mathcal{L}/\partial\beta_c\partial\beta_c < 0$ and that $\partial^2\mathcal{L}/\partial\beta_c\partial\beta_b > 0$. By definition of ψ, for each x, $(\partial\mathcal{L}/\partial\beta_c)(x, \psi(x)) = 0$. Since $\partial^2\mathcal{L}/\partial\beta_c\partial\beta_b > 0$, the function $\partial\mathcal{L}/\partial\beta_c$ is increasing in its first argument, and so $(\partial\mathcal{L}/\partial\beta_c)(x + \Delta, \psi(x)) > 0$ for all $\Delta > 0$. Since $\partial^2\mathcal{L}/\partial\beta_c\partial\beta_c < 0$, the function $\partial\mathcal{L}/\partial\beta_c$ is decreasing in its second argument, and hence $\psi(x + \Delta) > \psi(x)$, as desired.

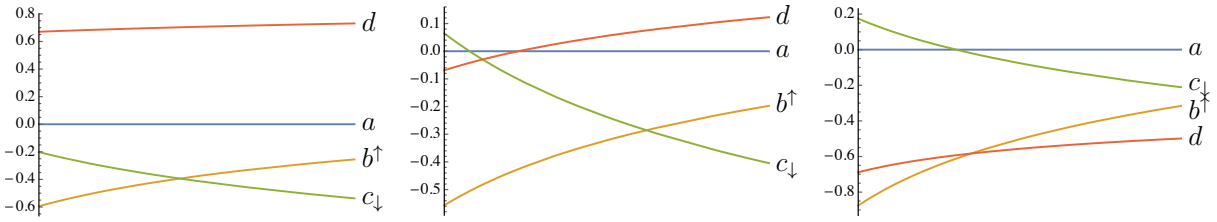


Figure 6.1: The MLE for Thurstone–Mosteller models is monotonic: with more $b \succ c$ comparisons, b's utility increases, while c's decreases. The vector shows the dataset # with $\mathcal{X}^2$ in lexic order.

6.5 Pairwise Majority Consistency

Social choice theory has its root in the analysis of politics, where in many cases it is important to use aggregation rules that respect the wishes of a majority. A famous issue is that the "majority will" may not be coherent and in particular fail to be transitive. A minimal majoritarian requirement, thus, would be what we call *pairwise majority consistency (PMC)*: in cases where the majority produces a definite ranking, the aggregate should respect it.

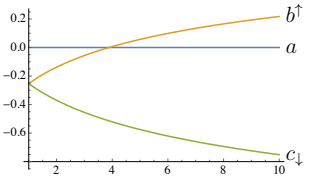
Definition 6.5.1. Suppose it is possible to label alternatives as $\mathcal{X} = \{x_1, \dots, x_m\}$ such that whenever $i < j$, it holds that $\#\{x_i \succ x_j\} > \#\{x_j \succ x_i\}$. Then, pairwise majority consistency (PMC) requires that for every MLE $\hat{\beta}$, it holds that $\hat{\beta}_{x_i} \geq \hat{\beta}_{x_j}$ for all $i < j$.

In contrast to our previous properties, PMC is violated by random utility models.

Example 6.5.2. *Consider $\mathcal{X} = \{a, b, c\}$, and consider the dataset*

$$\#\{a \succ b\} = 3, \#\{b \succ a\} = 2, \#\{a \succ c\} = 3, \#\{c \succ a\} = 2, \#\{b \succ c\} = 10, \#\{c \succ b\} = 1.$$

This dataset conforms to Definition 6.5.1 if we label $x_1, x_2, x_3 = a, b, c$. However, the unique MLE in the Thurstone–Mosteller model is $\hat{\beta}_a = 0$, $\hat{\beta}_b \approx 0.217$ and $\hat{\beta}_c \approx -0.751$, so that $\hat{\beta}_b > \hat{\beta}_a > \hat{\beta}_c$. The same example works for Bradley–Terry, which has MLE $\hat{\beta}_a = 0$, $\hat{\beta}_b \approx 0.316$ and $\hat{\beta}_c \approx -1.256$.



Why does the MLE not respect the majority ordering on this example? If the number $\#\{b \succ c\}$ was slightly above 1, we would obtain an MLE respecting the majority ordering, with $a \succ b \succ c$. However, as $\#\{b \succ c\}$ increases, due to the monotonicity of MLEs (Theorem 6.4.2), we find that $\hat{\beta}_b$ increases and $\hat{\beta}_c$ decreases. When $\#\{b \succ c\}$ becomes sufficiently large, $\hat{\beta}_b$ crosses $\hat{\beta}_a$. Thus, we find that the MLE has the ordering $b \succ a \succ c$, which violates PMC. The figure on the right shows this behavior in the style of Figure 6.1, as $\#\{b \succ c\}$ increases from 1 to 10; we can see that PMC is violated from about 4.

This reasoning applies more generally to other random utility models beyond Thurstone–Mosteller, and we can construct similar counterexamples for a large class of such models; see Appendix E.5.

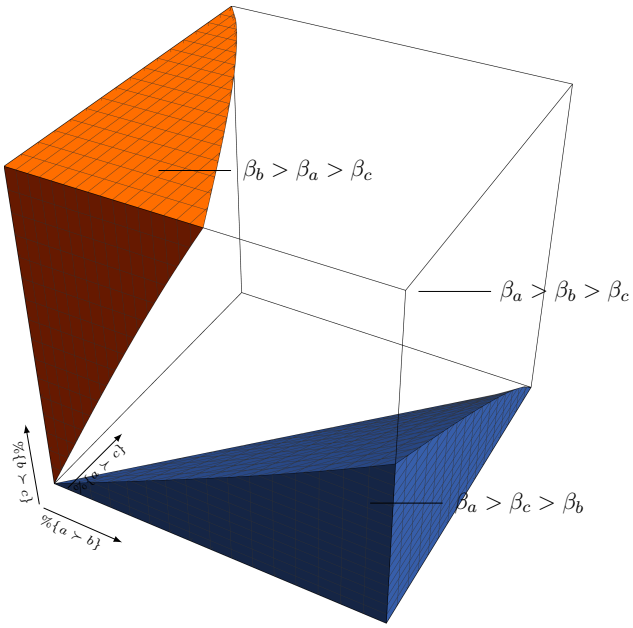


Figure 6.2: The cube shows all datasets in the space T, in which pairwise majority consistency requires that $\hat{\beta}_a > \hat{\beta}_b > \hat{\beta}_c$. The MLE for Thurstone-Mosteller models fails the condition in the shaded areas.

Theorem 6.5.3. *Maximum likelihood estimation violates pairwise majority consistency whenever F is strictly monotonic, strictly log-concave, and differentiable.*

How frequent are PMC violations? Write $\%\{x \succ y\} = \#\{x \succ y\}/(\#\{x \succ y\} + \#\{y \succ x\})$ for the fraction of x vs y comparisons that x wins. For $\mathcal{X} = \{a, b, c\}$, let T be the space of datasets with

$$0.5 < \%\{a \succ b\}, \%\{a \succ c\}, \%\{b \succ c\} \leq 1.$$

For all datasets in T, PMC requires that $\hat{\beta}_a > \hat{\beta}_b > \hat{\beta}_c$. In Figure 6.2, we draw the cube T and show the regions where the MLE for Thurstone–Mosteller fails PMC. Example 6.5.2, suitably normalized, falls in the upper orange region. Sampling uniformly over T, we find that Thurstone–Mosteller fails PMC in 17.8% of datasets, while Bradley–Terry fails in 16.6% of datasets.

6.6 Separability

We close by considering the *separability axiom* [Smi73; You75]. It requires that when we merge two datasets, then wherever the MLE agreed on the datasets, this agreement is preserved in the combined dataset.

Definition 6.6.1. Consider two datasets $\#^1$ and $\#^2$, and let $\hat{\beta}^1$ and $\hat{\beta}^2$ be MLEs. Suppose there exist two alternatives $a, b \in \mathcal{X}$ such that $\hat{\beta}^1_a > \hat{\beta}^1_b$ and $\hat{\beta}^2_a > \hat{\beta}^2_b$. Separability requires that for every MLE $\hat{\beta}$ for the pooled dataset $\# = \#^1 + \#^2$, it also holds that $\hat{\beta}_a > \hat{\beta}_b$.

Separability is also called *consistency*, and seems particularly desirable in cases where

we combine pairwise comparisons from different sources. While perhaps on first glance innocuous, separability is an extremely strong requirement, and few rules satisfy it; one can prove in general that separability constrains rules to be linear [Mye95]. Since likelihood maximization is not linear, it is no surprise that MLEs for random utility models fail separability.

Example 6.6.2. *Let $\mathcal{X} = \{a, b, c\}$, and consider the two datasets*

$$\#^1\{a \succ c\} = 6,\ \#^1\{c \succ a\} = 4,\ \#^1\{c \succ b\} = 100, \#^1\{b \succ c\} = 1, \text{ and}$$
$$\#^2\{a \succ c\} = 6,\ \#^2\{c \succ a\} = 4,\ \#^2\{b \succ a\} = 100, \#^2\{a \succ b\} = 1,$$

with 0 counts on all unspecified pairs. The unique MLEs for Thurstone–Mosteller on $\#^1$ and $\#^2$ are

$$\hat{\beta}^1_a = 0, \hat{\beta}^1_b \approx -2.58, \hat{\beta}^1_c \approx -0.253; \quad \text{and} \quad \hat{\beta}^2_a = 0, \hat{\beta}^2_b \approx 2.330, \hat{\beta}^2_c \approx -0.253.$$

We have both $\hat{\beta}^1_a > \hat{\beta}^1_c$ and $\hat{\beta}^2_a > \hat{\beta}^2_c$. However, the unique MLE on $\# = \#^1 + \#^2$ is $\hat{\beta}_a = 0$, $\hat{\beta}_b \approx 0.987$ and $\hat{\beta}_c \approx 1.973$. Thus. $\hat{\beta}_a < \hat{\beta}_c$, and so Thurstone–Mosteller violates separability. (The same example shows that Bradley–Terry violates separability.)

Intuitively, in both $\#_1$ and $\#_2$ there is a weak tendency to rank a above c, and the MLE can implement this tendency without incurring any cost on other pairs (since Lemma 6.2.2 applies). However, once we combine the datasets, a strong consensus for $c \succ b \succ a$ emerges, and overriding this consensus to ensure $a \succ c$ is not worth it. While failing separability, the MLE's behavior seems perfectly sensible, and we prove in Appendix E.6 that all random utility model do the same on this kind of example.

Theorem 6.6.3. *Maximum likelihood estimation violates separability whenever F is strictly monotonic, strictly log-concave, and differentiable.*

Like for PMC, we can again ask on what percentage of (pairs of) datasets the MLE fails separately. Since we sample over pairs, we might guess the answer to be of lower order than in the case of PMC, and this is borne out by the data. For $m = 3$ alternatives, sampling uniformly over the space of datasets for which each pair of distinct alternatives is compared equally often, we find that on about 1.5% of dataset pairs, Thurstone–Mosteller fails separability. This fraction increases as m increases, since there are more pairs of alternatives for which separability can be violated.

6.7 Discussion

To recap, we have established (under very mild assumptions) that the aggregation of pairwise comparisons via the MLE of a random utility model satisfies Pareto efficiency and monotonicity, and does not satisfy pairwise majority consistency and separability.

Our positive results deal with central properties that are required for an aggregation procedure: it does not override unanimous opinions (Pareto efficiency) and it incorporates new information (monotonicity). The latter property can be seen as a participation incentive, guaranteeing agents that each additional pairwise comparison will move the aggregate. Separability and pairwise majority consistency are not satisfied by random utility models,

but arguably these properties are not as universally desirable. An analogy to the world of ranking-based voting rules is instructive, where separability characterizes a specific class of aggregators (positional scoring rules) [You75], but none of them satisfies pairwise majority consistency [Mou83].

Overall, we view our results as lending normative support to — and a more nuanced understanding of — existing and future applications of random utilities models for societal decision making.

Part III

Reinforcement Learning

Chapter 7

Please be an Influencer? Contingency-Aware Influence Maximization

Most previous work on influence maximization in social networks assumes that the chosen influencers (or *seed nodes*) can be influenced with certainty (i.e., with no contingencies). In this chapter, we focus on using influence maximization in public health domains for assisting low-resource communities, where *contingencies* are common. It is very difficult in these domains to ensure that the seed nodes are influenced, as influencing them entails contacting/convincing them to attend training sessions, which may not always be possible. Unfortunately, previous state-of-the-art algorithms for influence maximization are unusable in this setting. This chapter tackles this challenge via the following four contributions: (i) we propose the Contingency Aware Influence Maximization problem and analyze it theoretically; (ii) we cast this problem as a Partially Observable Markov Decision Process and propose CAIMS (a novel POMDP planner) to solve it, which leverages a natural action space factorization associated with real-world social networks; and (iii) we provide extensive simulation results to compare CAIMS with existing state-of-the-art influence maximization algorithms. Finally, (iv) we provide results from a real-world feasibility trial conducted to evaluate CAIMS, in which key influencers in homeless youth social networks were influenced in order to spread awareness about HIV.

7.1 Introduction

The influence maximization problem is an NP-Hard combinatorial optimization problem [KKT03], which deals with finding a set of K influential seed nodes in a social network to optimally spread influence in the network according to some pre-specified diffusion model. It is a practically relevant problem with numerous potential applications in the real world, especially in public health domains involving low-resource communities. For example, it has been used to prevent smoking among teenagers [VP07], and to promote healthier lifestyles among risky populations [Ric10]. Recently, influence maximization algorithms were used to spread awareness about HIV among homeless youth with great results [Yad+17].

Recently, several efficient algorithms have been proposed (and deployed in the real-world) to solve influence maximization problems [Bor+14; TXS14; Coh+14; Wil+17]. Most of these algorithms rely on the following key assumption: seed nodes can be influenced with certainty. Unfortunately, in most public health domains, this assumption does not hold as "influencing" seed nodes entails training them to be "*peer leaders*" [VP07]. For example, seed nodes promoting HIV awareness among homeless youth need to be trained so that they can communicate information about supposedly private issues in a safe manner [SZL15]. This issue of training seed nodes leads to two practical challenges. First, it may be difficult to contact seed nodes in a timely manner (e.g., contacting homeless youth is challenging since they rarely have fixed phone numbers, etc). Second, these seed nodes may decline to be influencers (e.g., they may decline to show up for training sessions). In this paper, we refer to these two events as contingencies in the influence maximization process.

Unsurprisingly, these contingencies result in a wastage of valuable time/money spent in unsuccessfully contacting/convincing the seed nodes to attend the training. Moreover, the resulting influence spread achieved is highly sub-optimal, as very few seed nodes actually attend the training session, which defeats the purpose of conducting these interventions. Clearly, contingencies in the influence maximization process need to be considered very carefully.

In this paper, we propose a principled approach to handle these inevitable contingencies via the following contributions. First, we introduce the **C**ontingency **A**ware **I**nfluence **M**aximization (or CAIM) problem to handle cases when seed nodes may be unavailable, and analyze it theoretically. The principled selection of alternate seed nodes in CAIM (when the most preferred seed nodes are not available) sets it apart from any other previous work in influence maximization, which mostly assumes that seed nodes are always available for activation. Second, we cast the CAIM problem as a Partially Observable Markov Decision Process (POMDP) and solve it using CAIMS (**CAIM S**olver), a novel POMDP planner which provides an adaptive policy which explicitly accounts for contingency occurrences. CAIMS is able to scale up to real-world network sizes by leveraging the community structure (present in most real-world networks) to factorize the action space of our original POMDP into several smaller community-sized action spaces. Further, it utilizes insights from social network literature to represent belief states in our POMDP in a compact, yet accurate manner using Markov networks. Our simulations show that CAIMS outperforms state-of-the-art influence maximization algorithms by $\sim$60%. Finally, we evaluate CAIMS's usability in the real-world by using it to train a small set of homeless youth (the seed nodes) to spread awareness about HIV among their peers. This domain is an excellent testbed for CAIMS, as the transient nature of homeless youth increases the likelihood of the occurrence of contingencies [RR13].

7.2 Related Work

In addition to the work on influence maximization highlighted in the introduction, [Sin12] is related to our work as it solves an orthogonal problem: how to incentivize people in order to be influencers? Unlike us, they solve a mechanism-design problem where nodes have

private costs, which need to be paid for them to be influencers. However, in our domains of interest, monetary gains/losses are not the reason behind nodes getting influenced or not. Instead, nodes do not get influenced because of contingencies.

We also discuss work in POMDP planning, since we cast CAIM as a POMDP. SARSOP [KHL08] is a state-of-the-art offline POMDP solver but it does not scale up to larger state spaces. [SV10] proposed POMCP which use Monte-Carlo tree search in online planning, but it does not scale up to larger action spaces. As a result, FV-POMCP [AO15; SOA17] was proposed which relies on a factorized action space to scale up to larger problems. In our work, we complement their advances to build CAIMS, which leverages insights from social network theory to factorize action spaces in a provably "*lossless*" manner, and to represent beliefs in an accurate manner.

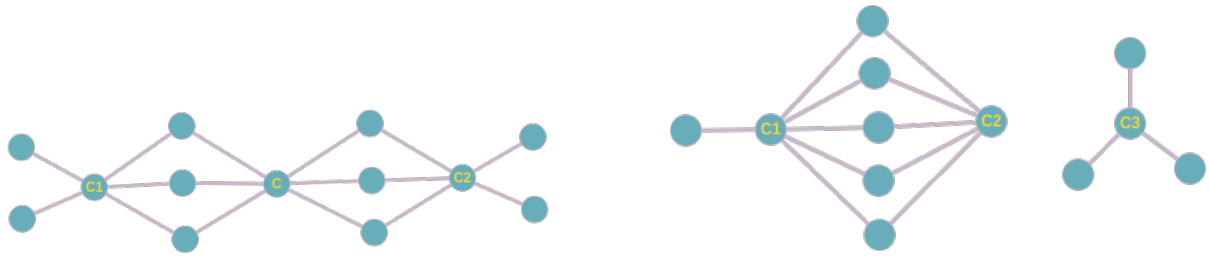
7.3 CAIM Model & Problem

We motivate our discussion of the CAIM problem by focusing on a particular public health domain: preventing HIV spread among homeless youth. In this domain, youth are highly susceptible to HIV infection due to high-risk activities that they engage in, e.g., unprotected sex, etc. [CDC13]. To reduce the spread of HIV, non-profit agencies called "homeless shelters" conduct intervention training camps to train influential homeless youth as "peer leaders", so that they can spread awareness about HIV in the *friendship based social network* of homeless youth, via peers in their social circles [Ric10].

Unfortunately, homeless shelters do not have the resources to train all homeless youth in the social network as peer leaders. Moreover, as behavioral problems of homeless youth makes managing larger groups difficult [Ric+12a], intervention training camps (interventions for short) can only include a small number ($\sim$5-6) of youth.

In practice, the shelter officials typically only have 4-5 days to locate/invite the desired youth to be trained. However, the transient nature of homeless youth (i.e., no fixed postal address, phone number, etc) makes contacting the chosen peer leaders difficult for homeless shelters. Further, most youth are distrustful of adults, and thus, they may decline to be trained as peer leaders [Mil+09]. As a result of these "*contingencies*", the shelter officials are often forced to conduct their intervention with very few peer leaders in attendance, despite each official spending 4-5 days worth of man hours in trying to find the chosen peer leaders [Yad+17]. Moreover, the peer leaders who finally attend the intervention are usually not influential seed nodes. This has been the state of operations even though peer-led interventions have been conducted by social workers for almost a decade now.

To avoid this outcome, ad-hoc measures have been proposed [Yad+17], e.g., contacting many more homeless youth than they can safely manage in an intervention. However, one then runs the risk that lots of youth may agree to be peer leaders, and shelter officials would have to conduct the intervention with all these youth (since it's unethical to invite a youth first and then ask him/her not to come to the intervention), even if the total number of such participants exceeds their maximum capacity [Ric+12b]. This results in interventions where the peer leaders may not be well trained, as insufficient attention is given to any one youth in the training. Note that if contingencies occurred infrequently, then inviting a



(a) Social Network 1 (b) Social Network 2

Figure 7.1: Examples illustrating harm in overprovisioning

few extra nodes (over the maximum capacity) may be a reasonable solution. However, as we show in the real-world feasibility trial conducted by us, contingencies are very common (*∼80%, or 14 out of 18 invitations in the real-world study resulted in contingencies*), and thus, overprovisioning by a small number of nodes is not an option. An ad-hoc fix for this over-attendance, is to first select (say) twice the desired number of homeless youth, invite them one at a time, and stop as soon as the desired number of homeless youth have accepted the invitation. However, we will show that this intuitive ad-hoc overprovisioning based solution performs poorly. First, we describe our influence model, followed by faults with overprovisioning.

Influence Model We represent friendship based social networks as undirected graphs $G = (\boldsymbol{V}, \boldsymbol{E})$, where each *node* $v \in \boldsymbol{V}$ represents a person in the social network and an *edge* $e = (A, B) \in \boldsymbol{E}$ between two nodes A and B (say) represents that nodes A and B are friends. Each edge $e \in \boldsymbol{E}$ has a propagation probability $p(e)$ associated with it, which represents the probability that a node which is influenced (has information) will pass on that influence to their neighbor. Influence spreads using the independent cascade model [KKT03], in which all nodes that get influenced at time t get a **single** chance to influence their un-influenced neighbors at time $t + 1$. This graph G with all relevant $p(e)$ values represents a *friendship based social network* and serves as an input to the CAIM problem.

Overprovisioning May Backfire Let K denote the number of nodes (or homeless youth) we want at the intervention. Now, suppose we overprovision by a factor of 2 and use the algorithm mentioned before. This means that instead of searching for the optimal set of K seed nodes, the algorithm finds the optimal set of $2K$ seed nodes and then influences the *first* K of these nodes that accept the invitation. Naturally, this algorithm should perform better (under contingencies) than the algorithm without overprovisioning. Surprisingly, we show that overprovisioning may make things worse. This happens because of two key ideas: (i) No K-sized subset of the optimal set of $2K$ nodes may be as good as the optimal set of K nodes (this indicates that we may not be looking for the right nodes when we search for the optimal set of $2K$ nodes), and (ii) An arbitrary K-sized subset of the optimal set of $2K$ nodes (obtained because we stick to the *first* K nodes that accept the invitation) may perform arbitrarily bad.

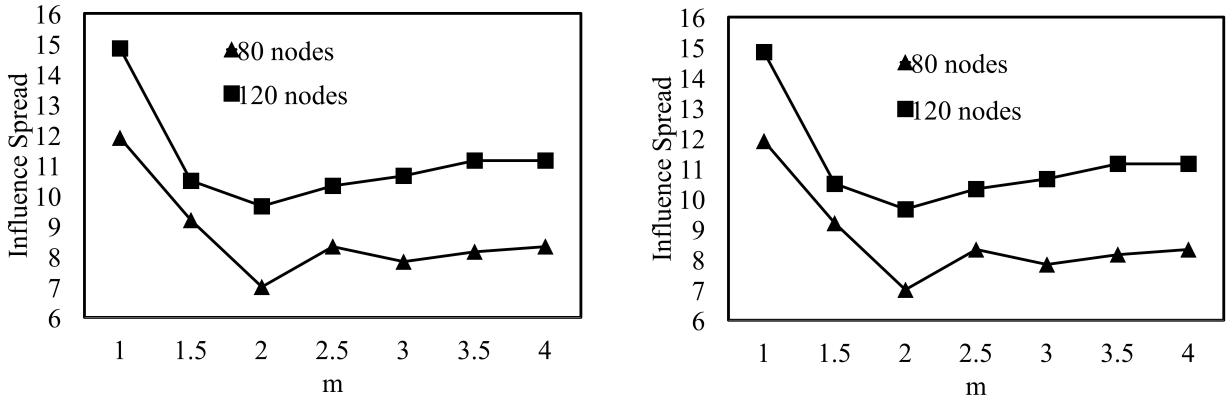
We now provide two examples that concretize these facts. For simplicity of the examples, we assume that influence spreads only for one round, number of nodes required for the

intervention is $K = 1$ and the propagation probability $p(e)$ is 0.5 for every edge. We use $\mathcal{I}(S)$ to denote the expected influence in the network when nodes of set S are influenced. Firstly, consider the example social network graph in Figure 7.1a. Suppose C and $C1$ are nodes that are regularly available, and are likely to accept the invitation. Now, let's find the best single node to influence for maximum influence spread. We don't need to consider nodes other than $\{C1, C, C2\}$ since they're obviously suboptimal. For the remaining nodes, we have $\mathcal{I}(C1) = 5 * 0.5 = 2.5$, $\mathcal{I}(C) = 6 * 0.5 = 3$ and $\mathcal{I}(C2) = 5 * 0.5 = 2.5$, and so the best single node to influence is C. Now, suppose we overprovision by a factor of 2, and try to find the optimal set of 2 nodes for maximum influence spread. The influence values are $\mathcal{I}(\{C1, C\}) = \mathcal{I}(\{C2, C\}) = 5 * 0.5 + 3 * 0.75 = 4.75$ and $\mathcal{I}(\{C1, C2\}) = 10 * 0.5 = 5$. So, the optimal set of 2 nodes to influence is $\{C1, C2\}$. But, since we need only one node, we would eventually be influencing either $C1$ or $C2$, giving us an expected influence of 2.5. On the other hand, if we did not overprovision, we would go for node C (the best single node to influence) and have an expected influence of 3. This example demonstrates that no K-sized subset of the optimal set of $2K$ nodes may be as good as the optimal set of K nodes. Note that, for clarity, the example considered here was small and made simple, and hence the difference between 3 and 2.5 may seem small. But, the example can be extended such that the difference is arbitrarily larger.

Secondly, consider the example social network graph of Figure 7.1b. Again, for simplicity, we assume that influence spreads only for one round, number of nodes required for the intervention is $K = 1$ and the propagation probability $p(e)$ is 0.5 for every edge. Like before, let's find the best single node to influence for maximum influence spread. We don't need to consider nodes other than $\{C1, C2, C3\}$ since they're obviously suboptimal. For the remaining nodes, we have $\mathcal{I}(C1) = 6 * 0.5 = 3$, $\mathcal{I}(C2) = 5 * 0.5 = 2.5$ and $\mathcal{I}(C3) = 3 * 0.5 = 1.5$, and so the best single node to influence is $C1$. Now, suppose we overprovision by a factor of 2, and try to find the optimal set of 2 nodes for maximum influence spread. The influence values are $\mathcal{I}(\{C1, C2\}) = 1 * 0.5 + 5 * 0.75 = 4.25$, $\mathcal{I}(\{C2, C3\}) = 8 * 0.5 = 4$ and $\mathcal{I}(\{C1, C3\}) = 9 * 0.5 = 4.5$. So, the optimal set of 2 nodes is $\{C1, C3\}$, and it would be selected by the overprovisioning algorithm. But, as mentioned before, we stop once we find the first node that accepts the invitation. Therefore, in case $C1$ is the first node encountered and it accepts the invitation, then there's an expected influence of 3, but if $C3$ is the first such node, the expected influence would be as low as 1.5. On the other hand, the standard algorithm (without overprovisioning) would directly go for $C1$ giving an expected influence of 3.

On a different note, suppose in this second example, node $C1$ is unavailable (because say it declines the invitation). In this case, the overprovisioning algorithm would have to go for $C3$ (the only other node in the optimal set of 2 nodes), leading to an expected influence of 1.5. However, an adaptive solution, would look for node $C1$ and after finding that its unavailable, would go for the next best node which is node $C2$. This gives an adaptive solution an expected influence of 2.5.

Having provided examples which provide intuition as to why simple ad-hoc overprovisioning based algorithms may backfire, we now provide empirical support for this intuition by measuring the performance of the Greedy algorithm [KKT03] (the gold standard in influence maximization) under varying levels of overprovisioning. Figures 7.2a and 7.2b



(a) SBM Networks (b) PA Networks

Figure 7.2: The Harm in Overprovisioning

compare influence spread achieved by Greedy on stochastic block model (SBM) and preferential attachment (PA) networks [SKP12], respectively, as it finds the optimal set of $m * K$ nodes ($K = 2$) to invite (i.e., overprovision by factor m) and influence the first K nodes that accept the invitation (the order in which nodes are invited is picked uniformly at random). The x-axis shows increasing m values and the y-axis shows influence spread. This figure shows that in both SBM and PA networks of different sizes, overprovisioning hurts, i.e., optimizing for larger seed sets in anticipation of contingencies actually hurts influence spread, which confirms our intuition outlined above. Overprovisioning's poor performance reveals that simple solutions do not work, thereby necessitating careful modeling of contingencies, as we do in CAIM.

Problem Setup Given a *friendship based social network*, the goal in CAIM is to invite several network nodes for the intervention until we get K nodes who agree to attend the intervention. The problem proceeds in T sequential sessions, where T represents the number of days that are spent in trying to invite network nodes for the intervention. In each session, we assume that nodes are either available or unavailable for invitation. This is because on any given day (session), homeless youth may either be present at the shelter (i.e., available) or not (i.e., unavailable). We assume that only nodes which are available in a given session can accept invitations in that session. This is because homeless youth frequently visit shelters, hence we utilize this opportunity to issue invitations to them if we see them at the shelter.

Let $\phi^t \in \{0,1\}^N$ (called a realization) be a binary vector which denotes the availability or unavailability (for invitation) of each network node in session $t \in [1,T]$. We take a Bayesian approach and assume that there is a known prior probability distribution $\mathbf{\Phi}$ over realizations ϕ^t such that $p(\phi^t) := \mathcal{P}[\mathbf{\Phi} = \phi^t]$. In our domain, this prior distribution is represented using a Markov Network. We assume that the realization ϕ^t for each session $t \in [1,T]$ is drawn i.i.d. from the prior distribution $\mathbf{\Phi}$, i.e., the presence/absence of homeless youth at the shelter in every session $t \in [1,T]$ is assumed to be an i.i.d. sample from $\mathbf{\Phi}$. We further assume that while the prior distribution $\mathbf{\Phi}$ is provided to the CAIM problem as input, the complete i.i.d. draws from this distribution (i.e., the realizations $\phi^t \; \forall t \in [1,T]$)

are not observable. This is because while querying the availability of a small number of nodes (∼3-4) is feasible, querying each node in the social network (which can have 150-160 nodes) for each session/day (to completely observe ϕ^t) requires a lot of work which is not possible with the shelters limited resources [Ric10].

In each session $t \in [1, T]$, a maximum of L actions can be taken, each of which can be of three possible types: *queries*, *invites* and *end-session* actions. Query action $q_{\boldsymbol{a}}$ in session $t \in [1, T]$ ascertains the availability/unavailability of a subset of nodes $\boldsymbol{a}$ ($\|\boldsymbol{a}\| \leqslant Q_{max}$, the maximum query size) in session t with certainty. Thus, query actions in session t provide partial *observations* about the realization of nodes ϕ^t in session t. On the other hand, invite action $m_{\boldsymbol{a}}$ invites a subset of nodes $\boldsymbol{a} \subset V$ ($\|\boldsymbol{a}\| \leqslant K$) to the intervention. Upon taking an invite action, we *observe* which invited nodes are present (according to ϕ^t) in the session and which of them accepted our invitation. Each invited node that is present accepts the invitation with a probability ϵ. We refer to the nodes that accept our invitation as "*locked* nodes" (since they are guaranteed to attend the intervention). Finally, we can also take an *end-session* action, if we choose not to invite/query any more nodes in that session.

The observations received from query and invite actions (*end-session* action provides no observation) taken in a session allows us to update the original prior distribution $\boldsymbol{\Phi}$ to generate a posterior distribution $\boldsymbol{\Phi}_t^{pos}(i)$ $\forall i \in [0, L]$ for session t (where i actions have been taken in session t so far). These posteriors can then be used to decide future actions that need to be taken in a session. Note that for every session t, $\boldsymbol{\Phi}_t^{pos}(0) = \boldsymbol{\Phi}$, i.e., at the beginning of each session, we start from the original prior distribution Φ and then get new posteriors every time we take an action in the session.

Note that even though query actions provide strictly lesser information than invite actions (for the same subset of nodes), their importance in CAIM is highlighted as follows: recall that the optimal set of 2 nodes in Figure 7.1b is $\{C1, C3\}$. If we remove the ability to query, we would invite nodes $C1$ and $C3$. In case $C1$ is not present and $C3$ accepts our invitation, we would be stuck with conducting intervention with only node $C3$ (since invited nodes who accept the invitation cannot be un-invited). Thus, we realize that inviting $C3$ is desirable only if $C1$ is present and accepts our invitation. Query actions allow us to query the presence or absence of both nodes $C1$ and $C3$ (so that we don't waste an invite action in case node $C1$ is found to be not present according to the query action's observation).

Informally then, given a friendship based social network $G = (\boldsymbol{V}, \boldsymbol{E})$, the integers T, K, L, Q_{max} and ϵ, and prior distribution $\boldsymbol{\Phi}$, the goal of CAIM is to find a policy for choosing L sequential actions for T sessions s.t. the expected influence spread (according to our influence model) achieved by the set of locked nodes (i.e., nodes which finally attend the intervention) is maximized.

Let $\boldsymbol{\mathcal{Q}} = \{q_{\boldsymbol{a}} \text{ s.t. } 1 \leqslant \|\boldsymbol{a}\| \leqslant Q_{max}\}$ denote the set of all possible query actions that can be taken in any given session $t \in [1, T]$. Similarly, let $\boldsymbol{\mathcal{M}} = \{m_{\boldsymbol{a}} \text{ s.t. } 1 \leqslant \|\boldsymbol{a}\| \leqslant K\}$ denote the set of all possible invite actions that can be taken in any given session $t \in [1, T]$. Also, let $\boldsymbol{\mathcal{E}}$ denote the end-session action. Let $\mathcal{A}_i^t \in \boldsymbol{\mathcal{Q}} \cup \boldsymbol{\mathcal{M}} \cup \boldsymbol{\mathcal{E}}$ denote the i^{th} action ($i \in [1, L]$) chosen by CAIM's policy in session $t \in [1, T]$.

Upon taking action $\mathcal{A}_i^t$ ($i \in [1, L], t \in [1, T]$), we receive observations which allow us to generate posterior distribution $\boldsymbol{\Phi}_t^{pos}(i)$. Denote by $\boldsymbol{M}_i^t$ the set of all locked nodes after the i^{th} action is executed in session t. Denote by $\boldsymbol{\Delta}$ the set of all possible posterior

distributions that we can obtain during the CAIM problem. Denote by $\mathbf{\Gamma}$ all possible sets of locked nodes that we can obtain during the CAIM problem. Finally, we define CAIM's policy $\mathbf{\Pi} : \mathbf{\Delta} \times \mathbf{\Gamma} \times [0, L] \times [1, T] \rightarrow \boldsymbol{\mathcal{Q}} \cup \boldsymbol{\mathcal{M}} \cup \boldsymbol{\mathcal{E}}$ as a function that takes in a posterior distribution, a set of locked nodes, the number of actions taken so far in the current session, and the session-id as input, and outputs an action $\mathcal{A}_i^t$ for the current timestep.

Problem 7.3.1. ***CAIM Problem*** *Given as input a social network $G = (\boldsymbol{V}, \boldsymbol{E})$ and integers T, K, L, Q_{max} and ϵ, and a prior distribution $\mathbf{\Phi}$ (as defined above), denote by $\mathcal{R}(\boldsymbol{M}_L^T)$ the expected total influence spread (i.e., number of nodes influenced) achieved by nodes in $\boldsymbol{M}_L^T$ (i.e., locked nodes at the end of T sessions). Let $\mathbb{E}_{\boldsymbol{M}_L^T \sim \mathbf{\Pi}}[\mathcal{R}(\boldsymbol{M}_L^T)]$ denote the expectation over the random variable $\boldsymbol{M}_L^T$, where $\boldsymbol{M}_L^T$ is updated according to actions recommended by policy $\mathbf{\Pi}(\mathbf{\Phi}_T^{pos}(L-1), \boldsymbol{M}_{L-1}^T, L-1, T)$. More generally, in session $t \in [1, T]$, $\boldsymbol{M}_i^t \ \forall i \in [0, L]$ is updated according to actions recommended by policy $\mathbf{\Pi}(\mathbf{\Phi}_t^{pos}(i-1), \boldsymbol{M}_{i-1}^t, i-1, t)$. Then, the objective of CAIM is to find an optimal policy $\mathbf{\Pi}^* = argmax_{\Pi} \mathbb{E}_{\boldsymbol{M}_L^T \sim \mathbf{\Pi}}[\mathcal{R}(\boldsymbol{M}_L^T)]$.*

We now theoretically analyze the CAIM problem. Some proofs are in Appendix F.

Lemma 7.3.2. *The CAIM problem is NP-Hard.*

Some NP-Hard problems exhibit nice properties that enable approximation guarantees for them. [GK11] introduced adaptive submodularity, the presence of which would ensure that a simple greedy algorithm provides a $(1 - 1/e)$ approximation w.r.t. the optimal CAIM policy. However, we show that while CAIM can be cast into the adaptive stochastic optimization framework of [GK11], our objective function is not adaptive submodular, because of which their Greedy algorithm does not have a $(1-1/e)$ approximation guarantee.

Lemma 7.3.3. *The objective function of CAIM is not adaptive submodular.*

These theorems show that CAIM is a computationally hard problem and it is difficult to even obtain any good approximate solutions for it. In this paper, we model CAIM as a POMDP.

7.4 POMDP Model

We cast the CAIM problem using POMDPs [Put09], as the uncertainty about the realization of nodes ϕ^t is similar to partial state observability in POMDPs. Finally, actions (queries and invites) that are chosen for the current session depend on the actions that are taken in future sessions (for e.g., influencing node A might be really important, but he/she may not be available in session t, therefore invite actions in session t can focus on other nodes, and influencing node A can be left to future sessions). This suggests the need to do lookahead search, which is the main motivation behind solving a POMDP. We now explain how we map CAIM onto a POMDP.

States A POMDP state consists of four entities $s = \langle \phi, \boldsymbol{M}, numAct, sessID \rangle$. Here, $sessID \in [1, T]$ identifies the session we are in. Also, $numAct \in [0, L]$ determines the number of actions that have been taken so far in session $sessID$. $\boldsymbol{M}$ denotes the set of locked nodes so far (starting from the first session). Finally, ϕ is the node realization ϕ^{sessID} in session $sessID$. In our POMDP model, states with $sessID = T$ and $numAct = L$ are

terminal states, since they represent the end of all sessions.

Actions A POMDP action is a tuple $a = \langle \boldsymbol{S}, type \rangle$. Here, *type* is a symbolic character which determines whether a is a query action (i.e., $type = q$), an invite action (i.e., $type = i$) or an *end-session* action (i.e., $type = e$). Also, $\boldsymbol{S} \subset \boldsymbol{V}$ denotes the subset of nodes that is queried ($type = q$) or invited ($type = i$). If $type = q$, the size of subset $\|\boldsymbol{S}\| \in [1, Q_{max}]$. Similarly, if $type = i$, $\|\boldsymbol{S}\| \in [1, K]$. Finally, if $type = e$, subset $\boldsymbol{S}$ is empty.

Observations Upon taking a query action $a = \langle \boldsymbol{S}, q \rangle$ in state $s = \langle \phi, \boldsymbol{M}, numAct, sessID \rangle$, we receive an observation that is completely determined by state s. In particular, we receive the observation $o_q = \{\phi(v)\ \forall v \in \boldsymbol{S}\}$, i.e., the availability status of each node in $\boldsymbol{S}$. And, by taking an invite action $a = \langle \boldsymbol{S}, i \rangle$ in state $s = \langle \phi, \boldsymbol{M}, numAct, sessID \rangle$, we receive two kinds of observations. Let $\boldsymbol{\Gamma} = \{v \in \boldsymbol{S}$ s.t. $\phi(v) = 1\}$ denote the set of available nodes in *invited set* $\boldsymbol{S}$. First, we get observation $o_i^1 = \{\phi(v)\ \forall v \in \boldsymbol{S}\}$ which specifies the availability status of each node in invited set $\boldsymbol{S}$. We also get an observation $o_i^2 = \{b(v)\ \forall v \in \boldsymbol{\Gamma}\}$ for each available node $v \in \boldsymbol{\Gamma}$, which denotes whether node v accepted our invitation and joined the locked set of nodes ($b(v) = 1$) or not ($b(v) = 0$). Finally, the *end-session* action does not generate any observations.

Rewards We only get rewards when we reach terminal states $s' = \langle \phi, \boldsymbol{M}, numAct, sessID \rangle$ with $sessID = T$, $numAct = L$. The reward attained in terminal state s' is the expected influence spread (as per our influence model) achieved by influencing nodes in the locked set $\boldsymbol{M}$ of s'.

Transition And Observation Probabilities Due to our exponential sized state and action spaces, maintaining transition and observation probability matrices is not feasible. Hence, we follow the paradigm of large-scale online POMDP solvers [SV10] by using a generative model $\Lambda(s, a) \sim (s', o, r)$ of the transition and observation probabilities. This generative model allows generating on-the-fly samples from the exact distributions $T(s'|s, a)$ and $\Omega(o|a, s')$ at very low computational costs. In our generative model, the state undergoes transitions as follows. On taking a query action, we reach a state s' which is the same as s except that $s'.numAct = s.numAct+1$. On taking an invite action $\langle \boldsymbol{S}, i \rangle$, we reach s' which is the same as s except that $s'.numAct = s.numAct + 1$, and $s'.\boldsymbol{M}$ is $s.\boldsymbol{M}$ appended with nodes of $\boldsymbol{S}$ that accept the invitation. Note that binary vector ϕ stays unchanged in either case (since the session does not change). Finally, on taking the end-session action, we start a new session by transitioning to state s' s.t., $s'.numAct = 0$, $s'.sessID = s.sessID + 1$, $s'.\boldsymbol{M} = s.\boldsymbol{M}$ and $s'.\phi$ is resampled i.i.d. from the prior distribution $\boldsymbol{\Phi}$. Note that the components $\boldsymbol{M}$, $numAct$ and $sessID$ of a state are fully observable. The observations (obtained on taking any action) are deterministically obtained as given in the "Observations" sub-section given above.

Initial Belief State The prior distribution $\boldsymbol{\Phi}$, along with other completely observable state components (such as $sessID = 1$, $numAct = 0$, and an empty locked set $\boldsymbol{M} = \{\}$) forms our initial belief state.

7.5 CAIMS: CAIM Solver

Our POMDP algorithm is motivated by the design of FV-POMCP, a recent online POMDP algorithm [AO15]. Unfortunately, FV-POMCP has several limitations which make it unsuitable for solving the CAIM problem. Thus, we propose CAIMS, a Monte-Carlo (MC) sampling based online POMDP algorithm which makes key modifications to FV-POMCP, and solves the CAIM problem for real-world sized networks. Next, we provide a brief overview of POMCP, and its extension FV-POMCP.

POMCP POMCP [SV10] uses UCT based Monte-Carlo tree search (MCTS) [Bro+12] to solve POMDPs. At every stage, given the current belief state b, POMCP incrementally builds a UCT tree that contains statistics that serve as empirical estimators (via MC samples) for the POMDP Q-value function $Q(b,a) = R(b,a) + \sum_z P(z|b,a) max_{a'} Q(b',a')$. The algorithm avoids expensive belief updates by maintaining the belief at each UCT tree node as an unweighted particle filter (i.e., a collection of all states that were reached at that UCT tree node via MC samples). In each MC simulation, POMCP samples a start state from the belief at the root node of the UCT tree, and then samples a trajectory that first traverses the partially built UCT tree, adds a node to this tree if the end of the tree is reached before the desired horizon, and then performs a random rollout to get one MC sample estimate of $Q(b,a)$. Finally, this MC sample estimate of $Q(b,a)$ is propagated up the UCT tree to update Q-value statistics at nodes that were visited during this trajectory. Note that the UCT tree grows exponentially large with increasing state and action spaces. Thus, the search is directed to more promising areas of the search space by selecting actions at each tree node h according to the UCB1 rule [KS06], which is given by: $a = argmax_a \hat{Q}(b_h,a) + c\sqrt{log(N_h+1)/n_{ha}}$. Here, $\hat{Q}(b_h,a)$ represents the Q-value statistic (estimate) that is maintained at node h in the UCT tree. Also, N_h is the number of times node h is visited, and n_{ha} is the number of times action a has been chosen at tree node h (POMCP maintains statistics for N_h and $n_{ha} \forall a \in A$ at each tree node h). While POMCP handles large state spaces (using MC belief updates), it is unable to scale up to large action sizes (as the branching factor of the UCT tree blows up). We validate POMCP's poor scale-up performance in our experiments.

FV-POMCP FV-POMCP extends POMCP to deal with large action spaces. It assumes that the action space of the POMDP can be factorized into a set of ℓ factors, i.e., each action a can be decomposed into a set of sub-actions $a_l \forall l \in [1,\ell]$. Under this assumption, the value function of the original POMDP is decomposable into a set of overlapping factors. i.e., $Q(b,a) = \sum_{l \in [1,\ell]} \alpha_l Q_l(b,a_l)$, where α_l ($\forall l \in [1,\ell]$) are factor-specific weights. FV-POMCP maintains a single UCT tree (similar to standard POMCP), but it differs in the statistics that are maintained at each node of the UCT tree. Instead of maintaining $\hat{Q}(b_h,a)$ and n_{ha} statistics for every action in the global (unfactored) action space at tree node h, it maintains a set of statistics that estimates the values $\hat{Q}_l(b_h,a_l)$ and n_{ha_l} $\forall l \in [1,\ell]$.

Joint actions are selected by the UCB1 rule across all factored statistics, i.e., $a = argmax_a \sum_{l \in [1,\ell]} \hat{Q}_l(b_h,a_l) + c\sqrt{log(N_h+1)/n_{ha_l}}$. This maximization is efficiently done using variable elimination (VE) [GKP02], which exploits the action factorization appropriately.

Thus, FV-POMCP achieves scale-up by maintaining fewer statistics at each tree node h, and by using VE to find the maximizing joint action.

However, there are two limitations which makes FV-POMCP unsuitable for solving CAIM. First, the VE procedure used in FV-POMCP (as described above) may return an action (i.e., a set of nodes) which is infeasible in the CAIM problem (e.g., the action may have more than K nodes). We elaborate on this point later. Second, FV-POMCP uses unweighted particle filters to represent belief states, which becomes highly inaccurate with exponentially sized state spaces in CAIM. We address these limitations in CAIMS.

7.5.1 CAIMS

CAIMS is an online Monte-Carlo sampling based POMDP solver that uses UCT based Monte-Carlo tree search to solve the CAIM problem. Similar to FV-POMCP, CAIMS also exploits action factorization to scale up to large action spaces. We now explain CAIMS's action factorization.

Action Factorization Real world social networks generally exhibit a lot of community structure, i.e., these networks are composed of several tightly-knit communities (partitions), with very few edges going across these communities [SKP12]. This community structure dictates the action factorization in CAIMS. As stated before, the POMDP model has each action of the form $\langle \boldsymbol{S}, type \rangle$, where $\boldsymbol{S}$ is a subset of nodes (that are being queried or invited). This (sub)set $\boldsymbol{S}$ can be represented as a boolean vector $\vec{S}$ (denoting which nodes are included in the set). Let $Q_q(\vec{S})$ denote the Q-value of the query action $\langle \boldsymbol{S}, q \rangle$, $Q_i(\vec{S})$ denote the Q-value of the invite action $\langle \boldsymbol{S}, i \rangle$ and let Q_e denote the Q-value of the end-session action $\langle \{\}, e \rangle$. Now, suppose the real-world social network is partitioned into ℓ partitions (communities) $P_1, P_2, \cdots P_\ell$. Let $\vec{S}_{P_x}$ denote the sub-vector of $\vec{S}$ corresponding to the x^{th} partition. Then, the action factorization used is: $Q_q(\vec{S}) = \sum_{x=1}^{\ell} Q_q^{P_x}(\vec{S}_{P_x})$ for query actions and $Q_i(\vec{S}) = \sum_{x=1}^{\ell} Q_i^{P_x}(\vec{S}_{P_x})$ for invite actions.

Intuitively, $Q_i^{P_x}(\vec{S}_{P_x})$ can be seen as the Q-value of inviting only nodes given by $\vec{S}_{P_x}$ (and no other nodes). Now, if querying/inviting nodes of one partition has negligible effect/influence on the other partitions, then the Q-value of the overall invite action $\langle \boldsymbol{S}, i \rangle$ can be approximated by the sum of the Q-values of the sub-actions $\langle \boldsymbol{S_{P_x}}, i \rangle$. The same holds for query actions. We now show that this action factorization is appropriate for CAIM as it introduces *minimal* error into the influence spread calculations for stochastic block model (SBM) networks, which mimic many properties of real-world networks [SKP12]. Note that we consider a single round of influence spread (T=1) as empirical research by Goel, Watts, and Goldstein [GWG12] shows that influence usually does not spread beyond the first hops (T=1) in real-world social networks.

Theorem 7.5.1. *Let $\mathcal{I}(\boldsymbol{S})$ denote the expected influence in the whole network when nodes of set $\boldsymbol{S}$ are influenced, and we have one round of influence spread. For an SBM network with n nodes and parameters (p, q) that is partitioned into ℓ communities, the difference between the true and factored expected influences can be bounded as $\mathbb{E}\left[\max_{\boldsymbol{S}} \left|\mathcal{I}(\boldsymbol{S}) - \sum_{x=1}^{\ell} \mathcal{I}(S_{P_x})\right|\right] \leq qn^2\left(1 - \frac{1}{\ell}\right) p_m$, where $p_m = \max_{e \in E} p(e)$ is the maximum propagation probability. Note that the (outer) expectation is over the randomness in*

the SBM network model.

This action factorization allows maintaining separate Q-value statistics ($\hat{Q}^{P_x}_{type}(\vec{S}_{P_x})\ \forall type \in \{q, i, e\}$) for each factor (i.e., network community) at each node of the UCT tree maintained by CAIMS. However, upon running MC simulations in this UCT tree, we acquire samples of only Q_{type} (i.e., rewards of the joint *un-factored* actions). We learn factored estimates $Q^{P_x}_{type}$ from estimates Q_{type} of the *un-factored* actions by using mixture of experts optimization [AO15], i.e. we estimate the factors as $\hat{Q}^{P_x}_{type}(\vec{S}_{P_x}) = \alpha_{P_x}\mathbb{E}[Q_{type}(\vec{S})|\vec{S}_{P_x}]$, where this expectation is estimated by using the empirical mean. Please refer to [AO15] for more details. We now describe action selection in the UCT tree.

Action Selection At each node in the UCT tree, we use the UCB1 rule (over all factors) to find the best action. Let $n^q_{h\vec{S}_{P_x}}$ (or $n^i_{h\vec{S}_{P_x}}$) denote the number of times a query (or invite) action with sub-action $\vec{S}_{P_x}$ has been taken from node h of the UCT tree. Let N_h denote the number of times tree node h has been visited. The best query action to be taken is given as $\langle \boldsymbol{S_q}, q\rangle$, where $\vec{S}_q = argmax_{\|\vec{S}\|_1 \leq Q_{max}} \sum_{x=1}^{\ell} \hat{Q}^{P_x}_q(b_h, \vec{S}_{P_x}) + c\sqrt{log(N_h+1)/n^q_{h\vec{S}_{P_x}}}$. Similarly, the best invite action to be taken is given as $\langle \boldsymbol{S_i}, i\rangle$, where $\vec{S}_i = argmax_{\|\vec{S}\|_1 \leq K-|M|} \sum_{x=1}^{\ell} \hat{Q}^{P_x}_i(b_h, \vec{S}_{P_x}) + c\sqrt{log(N_h+1)/n^i_{h\vec{S}_{P_x}}}$ (where M is the set of locked nodes at tree node h). Let V_q and V_i denote the value attained at the maximizing query and invite actions, respectively. Finally, let V_e denote the value of the end-session action, i.e. $V_e = \hat{Q}_e + c\sqrt{log(N_h+1)/n^e_h}$ where n^e_h is the number of times the end-session action has been taken from tree node h. Then, the values V_q, V_i and V_e are compared and the action corresponding to $max(V_q, V_i, V_e)$ is chosen.

Improved VE Note that the UCB1 equations to find maximizing query/invite actions (as described above) are of the form $argmax_{\|\vec{a}\|_1 \leq z} \sum_{x=1}^{\ell} f_x(\vec{a}_x)$ (where $\vec{a} \in \{0,1\}^n$). Unfortunately, plain application of VE (like FV-POMCP) to this results in infeasible solutions which may violate the L-1 norm constraint. Thus, FV-POMCP's VE procedure may not produce feasible solutions for CAIM.

CAIMS addresses this limitation by using two adjustments. First, we incorporate this L-1 norm constraint as an additional factor in the objective function: $argmax_{\vec{a}\in\{0,1\}^n} \sum_{x=1}^{\ell} f_x(\vec{a}_x) + f_c(\vec{a})$. This constraint factor f_c's scope is all the n variables (as it represents a global constraint connecting actions selected across all factors), and hence it can be represented using a table of size $O(2^n)$ in VE. Unfortunately, the exponentially sized table of f_c eliminates any speed-up benefits that VE provides, as the induced width of the tree formed (on running VE) will be n, leading to a worst possible time-complexity of $O(2^n)$.

To resolve this, CAIMS leverages a key insight which allows VE to run efficiently even with the additional factor f_c. The key idea is that, if all variables of a community are eliminated at once, then both (i)f_c; and (ii) the factors derived from a combination of f_c and other community-specific factors during such elimination, can be represented very concisely (using just tables of size $z+1$ elements), instead of using tables of size $O(2^n)$. This fact is straightforward to see for the original constraint factor f_c (as f_c's table only depends on $\|\vec{a}\|_1$, it has value 0 if $\|\vec{a}\|_1 \leq z$ and $-\infty$ otherwise). However, it is not

obvious why this holds for derived factors, which need to maintain optimal assignments to community-specific variables, for every possible combination of *un-eliminated* variable values (thereby requiring $O(2^n)$ elements). However, it turns out that we can still represent the derived factors concisely. The key insight is that even for these derived factors, all variable assignments with the same L-1 norm have the same value (Lemma 7.5.2). This allows us to represent each of these derived factor as a table of only $z+1$ elements (as we need to store one unique value when the L-1 norm is at most z, and we use $-\infty$ otherwise).

Lemma 7.5.2. *Let $\psi_i(\vec{v})$ denote the i^{th} factor generated during CAIMS's VE. Then, $\psi_i(\vec{v}_1) = \psi_i(\vec{v}_2)$ if $\|v_1\|_1 = \|v_2\|_1$. Further $\psi_i(\vec{v}) = -\infty$ if $\|v\|_1 > z$.*

These compact representations allow CAIMS to efficiently run VE in time $\sum_{i=1}^{\ell} O\left(2^{s_i}\right)$ (s_i = size of i^{th} community) even after adding the global constraint factor f_c (Lemma 7.5.3). This is the best one can do, because any algorithm will have to look at all values of each community-specific factor in order to solve the problem.

Lemma 7.5.3. *CAIMS's VE has time-complexity $\sum_{i=1}^{\ell} O\left(2^{s_i}\right)$, where s_i is the size of the i^{th} factor (community). There exists no procedure with better time complexity.*

Markov Net Beliefs FV-POMCP uses unweighted particle filters to represent beliefs, i.e. a belief is represented by a collection of states (also known as particles), wherein each particle has an equal probability of being the true state. Unfortunately, due to CAIM's exponential state-space, this representation of beliefs becomes highly inaccurate which leads to losses in solution quality.

To address this limitation, CAIMS makes the following assumption: availability of network nodes is positively correlated with the availability of their neighboring nodes in the social network. This assumption is reasonable because homeless youth usually go to shelters with their friends [RR13]. Thus, the confirmed availability of one homeless youth increases the likelihood of the availability of his/her friends (and vice versa). Under this assumption, the belief state in CAIM can be represented using a Markov Network. Formally, the belief is given as $b = \langle \mathcal{N}, \boldsymbol{M}, numAct, sessID \rangle$, where $\mathcal{N}$ is a Markov Network representing our belief of the true realization ϕ (note that the other three components of a state are observable). With the help of this Markov Network, we maintain *exact* beliefs throughout the POMCP tree of CAIMS. As mentioned before, the prior distribution $\boldsymbol{\Phi}$ that serves as part of the initial belief state is also represented using a Markov Network $\mathcal{N}_0$. This prior can be elicited from field observations made by homeless shelter officials, and can be refined over multiple runs of CAIMS. *In our simulations, the social network structure $G = (\boldsymbol{V}, \boldsymbol{E})$ is used as a surrogate for the Markov network structure, i.e., the Markov network only has potentials over two variables/nodes (one potential for each pair of nodes connected by an edge in social network G).* Thus, we start with the initial belief as $\langle \mathcal{N}_0, \{\}, 0, 1 \rangle$. Upon taking actions $a = \langle S, type \rangle$ and receiving observations o, the belief state can be updated by conditioning the Markov network on the observed variables (i.e., by conditioning the presence/absence of nodes based on observations received from past query actions taken in the current session). This helps us maintain exact beliefs throughout the POMCP tree efficiently, which helps CAIMS take more accurate decisions.

7.6 Evaluation

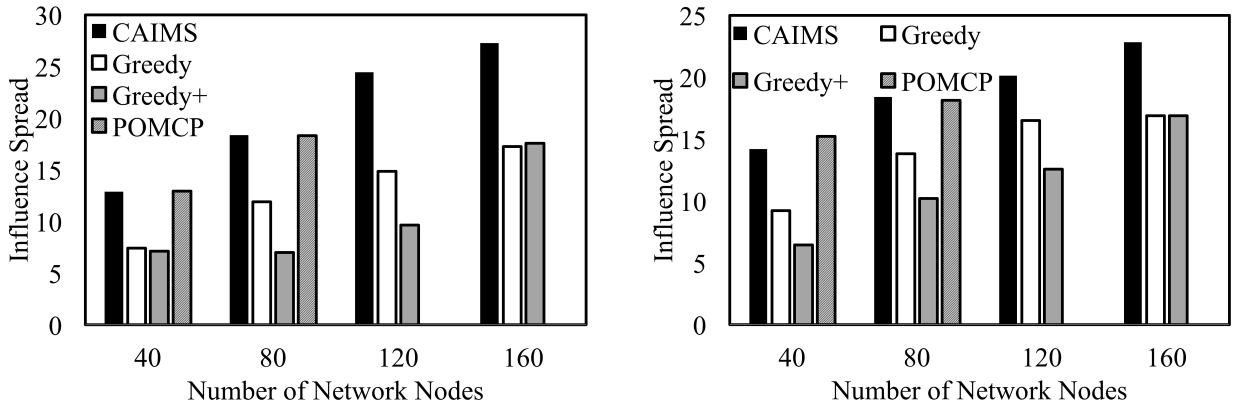
We show simulation results on artificially generated (and real-world) networks to validate CAIMS's performance in a variety of settings. We also provide results from a real-world feasibility study involving 54 homeless youth which shows the real-world usability of CAIMS. For our simulations, all the networks were generated using NetworkX library [HSS08]. All experiments are run on a 2.4 GHz 8-core Intel machine having 128 GB RAM. Unless otherwise stated, we set $L = 3$, $Q_{max} = 2$, $K = 2$, and all experiments are averaged over 50 runs. *All simulation results are statistically significant under t-test ($\alpha = 0.05$).*

Baselines We use two different kinds of baselines. For influence maximization solvers, we use Greedy [KKT03], the gold-standard in influence maximization as a benchmark. We subject Greedy's chosen nodes to contingencies drawn from the same prior $\mathbf{\Phi}$ distribution that CAIMS uses. We also compare against the overprovisioning variant of Greedy (Greedy+) where instead of selecting K nodes, we select $2K$ nodes and influence the first K nodes that accept the invitation. This was proposed as an ad-hoc solution in [Yad+17] to tackle contingencies, and hence, we compare CAIMS against this. We also compare CAIMS against state-of-the-art POMDP solvers such as SARSOP and POMCP. Unfortunately, FV-POMCP cannot be used for comparison as its VE procedure is not guaranteed to satisfy the K budget constraint used inside CAIMS.

Solution Quality Comparison Figures 7.3a, 7.3b and 7.6a compares influence spread of CAIMS, Greedy, Greedy+ and POMCP on SBM ($p = 0.4, q = 0.1$), Preferential Attachment (PA) ($n = 5$) and real-world homeless youth networks (used in [Yad+16]), respectively. We select $K = 2$ nodes, and set $T = 6, L = 3$ for CAIMS. The X-axis shows the size of the networks and the Y-axis shows the influence spread achieved. Figures 7.3a and 7.3b show that on SBM and PA networks, POMCP runs out of memory on networks of size 120 nodes. Further, these figures also show that CAIMS significantly outperforms Greedy and Greedy+ on both SBM (by ∼73%) and PA networks (by ∼58%). Figure 7.6a shows that even on real-world networks of homeless youth (which had ∼160 nodes each) , POMCP runs out of memory, while CAIMS outperforms Greedy and Greedy+ by ∼25%. This shows that state-of-the-art influence maximization solvers perform poorly in the presence of contingencies, and a POMDP based method (CAIMS) outperforms them by explicitly accounting for contingencies. Figures 7.3a and 7.3b also show that Greedy+ performs worse than Greedy.

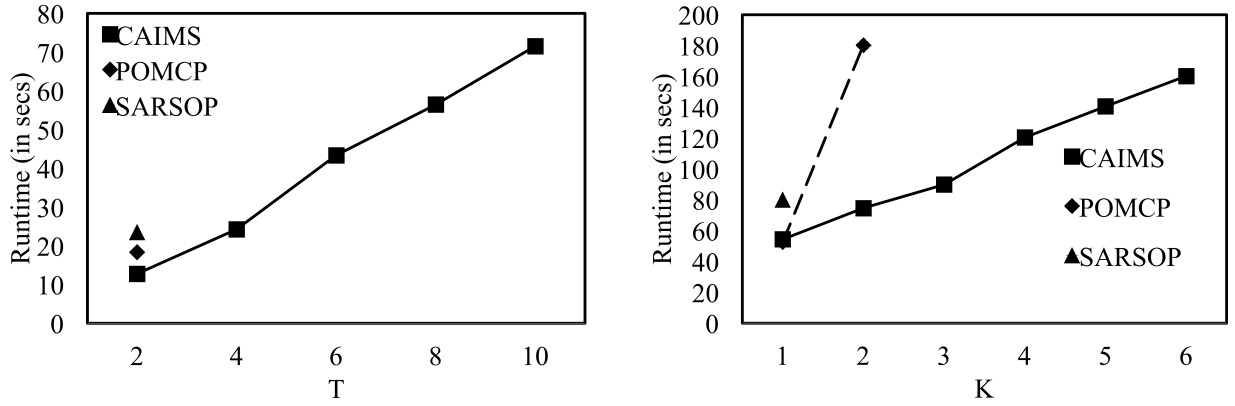
Scale up Having established the value of POMDP based methods, we now compare CAIMS's scale-up performance against other POMDP solvers. Figures 7.4a and 7.4b compares the runtime of CAIMS, POMCP and SARSOP on a 100 node SBM network with increasing values of T and K respectively. The X-axis shows T (or K) values and the Y-axis shows the influence spread. Figure 7.4a shows that both POMCP and SARSOP run out of memory at $T = 2$ sessions. On the other hand, CAIMS scales up gracefully to increasing number of sessions. Similarly, Figure 7.4b ($T = 10$) shows that SARSOP runs out of memory at $K = 1$, whereas POMCP runs out memory at $K = 2$, whereas CAIMS scales up to larger values of K. These figures show the superiority of CAIMS over its baselines as it outperforms them over a multitude of parameters and network classes.

Markov Nets We illustrate the value of Markov networks to represent belief states



(a) SBM Networks (b) PA Networks

Figure 7.3: Influence Spread Comparison

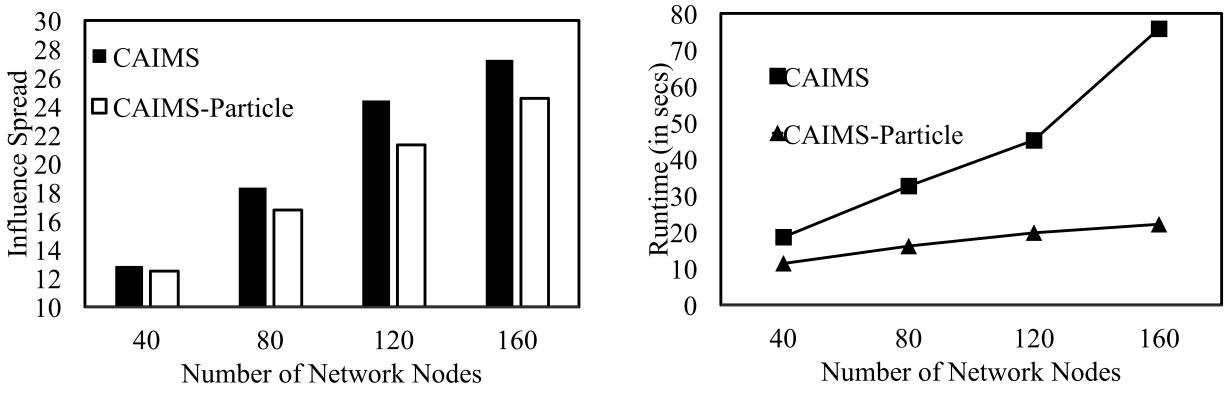


(a) Scale up in T (b) Scale up in K

Figure 7.4: Scale Up Results

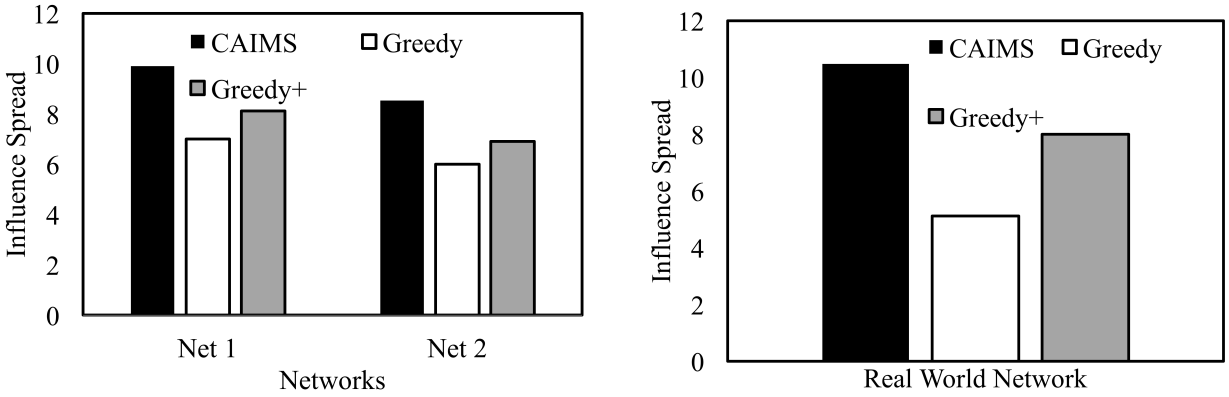
in CAIMS. We compare CAIMS with and without Markov nets (in this case, belief states are represented using unweighted particle filters) on SBM networks of increasing size. Figure 7.5a shows influence spread comparison between CAIMS and CAIMS-Particle (the version which uses unweighted particle filters to represent belief states). Figure 7.5b shows runtime comparison of CAIMS and CAIMS-Particle on the same SBM networks. These figures shows that using a more accurate representation for the belief state (using Markov networks) improved solution qualities by ∼15% at the cost of ∼3X slower runtime. However, the loss in speed due to Markov networks is not a concern (as even on 160 node networks, CAIMS with Markov networks runs in ∼75 seconds).

Real World Trial We conducted a real-world feasibility trial to test out CAIMS with a homeless shelter in Los Angeles. We enrolled 54 homeless youth from this shelter into our trial and constructed a friendship based social network for these youth (using social media contacts). The prior $\mathbf{\Phi}$ was constructed using field observations made by shelter officials. We then executed policies generated by CAIMS, Greedy and Greedy+ on this network ($K = 4$, $Q_{max} = 4$ and $L = 3$) on three successive days ($T = 3$) in the shelter to invite homeless



(a) Influence Spread (b) Runtime

Figure 7.5: Value of using Markov Networks



(a) Homeless Youth Networks (b) Feasibility Trial

Figure 7.6: Real World Experiments

youth to attend the intervention. In reality, 14 out of 18 invitations (∼80%) resulted in contingency events, which illustrates the importance of accounting for contingencies in influence maximization. Figure 7.6b compares influence spread (in simulation) achieved by nodes in invited sets selected by CAIMS, Greedy and Greedy+. This figure shows that CAIMS is able to spread 31% more influence as compared to Greedy and Greedy+.

7.7 Conclusion

Most previous influence maximization algorithms rely on the following assumption: seed nodes can be influenced with certainty. Unfortunately, this assumption does not hold in most real-world domains. This paper presents CAIMS, a contingency-aware influence maximization algorithm for selecting key influencers in a social network. Specifically, this paper makes the following five contributions: (i) we propose the Contingency-Aware Influence Maximization problem and provide a theoretical analysis of the same; (ii) we cast this problem as a Partially Observable Markov Decision Process (POMDP); (iii) we propose

CAIMS, a novel POMDP planner which leverages a natural action space factorization associated with real-world social networks; (iv) we provide extensive simulation results to compare CAIMS with existing state-of-the-art influence maximization algorithms; and (v) we test CAIMS in a real-world feasibility trial which confirms that CAIMS is indeed a usable algorithm in the real world.

Chapter 8

Teaching AI Agents Ethical Values Using Reinforcement Learning and Policy Orchestration

Autonomous cyber-physical agents play an increasingly large role in our lives. To ensure that they behave in ways aligned with the values of society, we must develop techniques that allow these agents to not only maximize their reward in an environment, but also to learn and follow the implicit constraints of society. We detail a novel approach that uses inverse reinforcement learning to learn a set of unspecified constraints from demonstrations and reinforcement learning to learn to maximize environmental rewards. A contextual bandit-based orchestrator then picks between the two policies: constraint-based and environment reward-based. The contextual bandit orchestrator allows the agent to mix policies in novel ways, taking the best actions from either a reward-maximizing or constrained policy. In addition, the orchestrator is transparent on which policy is being employed at each time step. We test our algorithms using Pac-Man and show that the agent is able to learn to act optimally, act within the demonstrated constraints, and mix these two functions in complex ways.

8.1 Introduction

Concerns about the ways in which autonomous decision making systems behave when deployed in the real world are growing. Stakeholders worry about systems achieving goals in ways that are not considered acceptable according to values and norms of the impacted community, also called "specification gaming" behaviors [RM19]. Thus, there is a growing need to understand how to constrain the actions of an AI system by providing boundaries within which the system must operate. To tackle this problem, we may take inspiration from humans, who often constrain the decisions and actions they take according to a number of exogenous priorities, be they moral, ethical, religious, or business values [Sen74; Lor+18a; Lor+18b], and we may want the systems we build to be restricted in their actions

by similar principles [AKS17]. The overriding concern is that the agents we construct may not obey these values while maximizing some objective function [Sim18; RM19].

The idea of teaching machines right from wrong has become an important research topic in both AI [Yu+18] and related fields [WA08]. Much of the research at the intersection of artificial intelligence and ethics falls under the heading of *machine ethics*, i.e., adding ethics and/or constraints to a particular system's decision making process [AA11]. One popular technique to handle these issues is called *value alignment*, i.e., restrict the behavior of an agent so that it can only pursue goals which follow values that are aligned to human values [RDT15; Lor+18b; Lor+18a].

Another important notion for these autonomous decision making systems is the idea of *transparency* or *interpretability*, i.e., being able to see why the system made the choices it did. Theodorou, Wortham, and Bryson [TWB16] observe that the Engineering and Physical Science Research Council (EPSRC) Principles of Robotics dictates the implementation of transparency in robotic systems. The authors go on to define transparency in a robotic or autonomous decision making system as "a mechanism to expose the decision making of the robot".

While giving a machine a code of morals or ethics is important, there is still the question of *how to provide the behavioral constraints to the agent*. A popular technique is called the *bottom-up approach*, i.e., teaching a machine what is right and wrong by example [ASW05; Bal+19a; Bal+18]. In this paper, we adopt this approach as we consider the case where only examples of the correct behavior are available to the agent, and it must therefore learn from only these examples.

We propose a framework which enables an agent to learn two policies: (1) π_R, a reward maximizing policy obtained through direct interaction with the world, and (2) π_C, obtained via inverse reinforcement learning over demonstrations by humans or other agents of how to obey a set of behavioral constraints in the domain. Our agent then uses a contextual-bandit-based orchestrator [BR19; Bou+17] to learn to blend the policies in a way that maximizes a convex combination of the rewards and constraints. Within the RL community this can be seen as a particular type of apprenticeship learning [AN04b] where the agent is learning how to be *safe*, rather than only maximizing reward [Lei+17].

One may argue that we should employ π_C for all decisions as it will be more 'safe' than employing π_R. Indeed, although one could use π_C exclusively for the agent, there are a number of reasons to employ the orchestrator. First, while the humans or other demonstrators may be good at demonstrating the constrained behavior, they may not provide good examples of how best to maximize reward. Second, the demonstrators may not be as creative as the agent when mixing the two policies [VG18]. By allowing the orchestrator to learn when to apply which policy, the agent may be able to devise better ways to blend the policies, leading to behavior which both follows the constraints and achieves higher reward than any of the human demonstrations. Third, we may not want to obtain demonstrations of what to do in all parts of the domain e.g., there may be dangerous or hard-to-model regions, or there may be mundane parts of the domain in which human demonstrations are too costly or boring to obtain. In this case, having the agent learn what to do in the non-demonstrated parts through RL is complementary. Finally, as we have argued, interpretability is an important feature to have. Although the policies

themselves may not be directly interpretable (though there is recent work in this area [Ver+18; Liu+18]), our proposed explicit orchestrator captures the notion of transparency and interpretability as we can see which policy is being applied in real time.

Contributions. We propose and test a novel approach to teach machines to act in ways that achieve and compromise multiple objectives in a given environment. One objective is the desired goal and the other one is a set of behavioral constraints, learnt from examples. Our technique uses aspects of both traditional reinforcement learning and inverse reinforcement learning to identify policies that both maximize rewards and follow particular constraints within an environment. Our agent then blends these policies in novel and interpretable ways using an orchestrator based on the contextual bandits framework. We demonstrate the effectiveness of these techniques on the Pac-Man domain where the agent is able to learn both a reward-maximizing and a constrained policy, and select between these policies in a transparent way based on context, to employ a policy that achieves high reward *and* obeys the demonstrated constraints.

8.2 Related Work

Ensuring that autonomous systems act in line with our values while achieving their objectives is a major research topic in AI. These topics have gained popularity among a broad community including philosophers [WA08] and non-profits [RDT15]. Yu, Shen, Miao, Leung, Lesser, and Yang [Yu+18] provide an overview of much of the recent research at major AI conferences on ethics in AI.

Agents may need to balance objectives and feedback from multiple sources when making decisions. One prominent example is the case of autonomous cars. There is extensive research from multidisciplinary groups into the questions of when autonomous cars should make lethal decisions [BSR16], how to aggregate societal preferences to make these decisions [Noo+18], and how to measure distances between these notions [Lor+18a; Lor+18b]. In a recommender systems setting, a parent or guardian may want the agent to not recommend certain types of movies to children, even if this recommendation could lead to a high reward [Bal+18; Bal+19a]. Recently, as a compliment to their concrete problems in AI saftey which includes reward hacking and unintended side effects [Amo+16], a DeepMind study has compiled a list of specification gaming examples, where very different agents game the given specification by behaving in unexpected (and undesired) ways.[1]

Within the field of reinforcement learning there has been specific work on ethical and interpretable RL. Wu and Lin [WL18] detail a system that is able to augment an existing RL system to behave ethically. In their framework, the assumption is that, given a set of examples, most of the examples follow ethical guidelines. The system updates the overall policy to obey the ethical guidelines learned from demonstrations using IRL. However, in this system only one policy is maintained so it has no transparency. Laroche and Feraud [LF17] introduce a system that is capable of selecting among a set of RL policies depending on context. They demonstrate an orchestrator that, given a set of policies for a particular

[1] 38 AI "specification gaming" examples are available at: https://docs.google.com/spreadsheets/d/e/2PACX-1vRPiprOaC3HsCf5Tuum8bRfzYUiKLRqJmbOoC-32JorNdfyTiRRsR7Ea5eWtvsWzuxo8bjOxCG84dAg/pubhtml

domain, is able to assign a policy to control the next episode. However, this approach use the classical multi-armed bandit, so the state context is not considered.

Interpretable RL has received significant attention in recent years. Luss and Petrik [LP16] introduce action constraints over states to enhance the interpretability of policies. Verma, Murali, Singh, Kohli, and Chaudhuri [Ver+18] present a reinforcement learning framework, called Programmatically Interpretable Reinforcement Learning (PIRL), that is designed to generate interpretable and verifiable agent policies. PIRL represents policies using a high-level, domain-specific programming language. Such programmatic policies have the benefit of being more easily interpreted than neural networks, and being amenable to verification by symbolic methods. Additionally, Liu, Schulte, Zhu, and Li [Liu+18] introduce Linear Model U-trees to approximate neural network predictions. An LMUT is learned using a novel on-line algorithm that is well-suited for an active play setting, where the mimic learner observes an ongoing interaction between the neural net and the environment. The transparent tree structure of an LMUT facilitates understanding the learned knowledge by analyzing feature influence, extracting rules, and highlighting the super-pixels in image inputs.

8.3 Background

8.3.1 Reinforcement Learning

Reinforcement learning defines a class of algorithms solving problems modeled as a Markov decision process (MDP) [SB98b]. An MDP is usually denoted by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where: $\mathcal{S}$ is a set of possible states; $\mathcal{A}$ is a set of actions; $\mathcal{T}$ is a transition function defined by $\mathcal{T}(s, a, s') = \Pr(s'|s, a)$, where $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is a reward function; γ is a discount factor that specifies how much long term reward is kept. The goal in an MDP is to maximize the discounted long term reward received. Usually the infinite-horizon objective is considered: $\max \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1})$.

Solutions come in the form of policies $\pi : \mathcal{S} \mapsto \mathcal{A}$, which specify what action the agent should take in any given state deterministically or stochastically. One way to solve this problem is through Q-learning with function approximation [BT96a]. The Q-value of a state-action pair, $\mathcal{Q}(s, a)$, is the expected future discounted reward for taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. A common method to handle very large state spaces is to approximate the $\mathcal{Q}$ function as a linear function of some features. Let $\boldsymbol{\psi}(s, a)$ denote relevant features of the state-action pair $\langle s, a \rangle$. Then, we assume $\mathcal{Q}(s, a) = \boldsymbol{\theta} \cdot \boldsymbol{\psi}(s, a)$, where $\boldsymbol{\theta}$ is an unknown vector to be learned by interacting with the environment. Every time the RL agent takes action a from state s, obtains immediate reward r, and reaches new state s', the parameter $\boldsymbol{\theta}$ is updated using

$$\begin{aligned} \text{difference} &= \left[r + \gamma \max_{a'} \mathcal{Q}(s', a') \right] - \mathcal{Q}(s, a) \\ \theta_i &\leftarrow \theta_i + \alpha \cdot \text{difference} \cdot \psi_i(s, a), \end{aligned} \tag{8.1}$$

where α is the learning rate. A common strategy used for exploration is ϵ-greedy: during the training phase, a random action is played with a probability of ϵ and the action with

maximum Q-value is played otherwise. The agent follows this strategy and updates the parameter $\boldsymbol{\theta}$ according to (8.1) until either the Q-values converge or a maximum number of time-steps is met.

8.3.2 Inverse Reinforcement Learning

IRL seeks to find the most likely reward function $\mathcal{R}_E$, which an expert E is executing [AN04b; NR00b]. IRL methods assume the presence of an expert that solves an MDP, where the MDP is fully known and observable by the learner except for the reward function. Since the state and action of the expert is fully observable by the learner, it has access to trajectories executed by the expert. A trajectory consists of a sequence of state and action pairs, $Tr = (s_0, a_0, s_1, a_1, \ldots, s_{L-1}, a_{L-1}, s_L)$, where s_t is the state of the environment at time t, a_t is the action played by the expert at the corresponding time and L is the length of this trajectory. The learner is given access to m such trajectories $\{Tr^{(1)}, Tr^{(2)}, \ldots, Tr^{(m)}\}$ to learn the reward function. Since the space of all possible reward functions is extremely large, it is common to represent the reward function as a linear combination of $\ell > 0$ features. $\widehat{\mathcal{R}}_{\boldsymbol{w}}(s, a, s') = \sum_{i=1}^{\ell} w_i \phi_i(s, a, s')$, where w_i are weights to be learned, and $\phi_i(s, a, s') \to \mathbb{R}$ is a feature function that maps a state-action-state tuple to a real value, denoting the value of a specific feature of this tuple. Current state-of-the-art IRL algorithms utilize feature expectations as a way of evaluating the quality of the learned reward function [SB17]. For a policy π, the feature expectations starting from state s_o are defined as

$$\mu(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}(s_t, a_t, s_{t+1}) \Big| \pi\right],$$

where the expectation is taken with respect to the state sequence achieved on taking actions according to π starting from s_0. One can compute an empirical estimate of the feature expectations of the expert's policy with the help of the trajectories $\{Tr^{(1)}, Tr^{(2)}, \ldots, Tr^{(m)}\}$, using

$$\hat{\mu}_E = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{L-1} \gamma^t \boldsymbol{\phi}\big(s_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)}\big). \tag{8.2}$$

Given a weight vector $\boldsymbol{w}$, one can compute the optimal policy $\pi_{\boldsymbol{w}}$ for the corresponding reward function $\widehat{\mathcal{R}}_{\boldsymbol{w}}$, and estimate its feature expectations $\hat{\mu}(\pi_{\boldsymbol{w}})$ in a way similar to (8.2). IRL compares this $\hat{\mu}(\pi_{\boldsymbol{w}})$ with expert's feature expectations $\hat{\mu}_E$ to learn best fitting weight vectors $\boldsymbol{w}$.

8.3.3 Contextual Bandits

Following Langford and Zhang [LZ08], the contextual bandit problem is defined as follows. At each time $t \in \{0, 1, \ldots, (T-1)\}$, the player is presented with a *context vector* $c(t) \in \mathbb{R}^d$ and must choose an arm $k \in [K] = \{1, 2, \ldots, K\}$. Let $\mathbf{r} = (r_1(t), \ldots, r_K(t))$ denote a reward vector, where $r_k(t)$ is the reward at time t associated with the arm $k \in [K]$. We assume

that the expected reward is a linear function of the context, i.e. $\mathbb{E}[r_k(t)|c(t)] = \mu_k^T c(t)$, where μ_k is an unknown weight vector (to be learned from the data) associated with the arm k.

The purpose of a contextual bandit algorithm A is to minimize the cumulative regret. Let $H : C \to [K]$ where C is the set of possible contexts and $c(t)$ is the context at time t, $h_t \in H$ a hypothesis computed by the algorithm A at time t and $h_t^* = \underset{h_t \in H}{\text{argmax}}\ r_{h_t(c(t))}(t)$ the optimal hypothesis at the same round. The cumulative regret is: $R(T) = \sum_{t=1}^{T} r_{h_t^*(c(t))}(t) - r_{h_t(c(t))}(t)$.

One widely used way to solve the contextual bandit problem is the Contextual Thompson Sampling algorithm (CTS) [AG13] given as Algorithm 8.1. In CTS, the

Algorithm 8.1: Contextual Thompson Sampling Algorithm

Initialize: $B_k = I_d$, $\hat{\mu}_k = 0_d$, $f_k = 0_d$ for $k \in [K]$.
for $t = 0, 1, 2, \ldots, (T-1)$ **do**
  Sample $\tilde{\mu}_k(t)$ from $N(\hat{\mu}_k, v^2 B_k^{-1})$.
  Play arm $k_t = \underset{k \in [K]}{argmax}\ c(t)^\top \tilde{\mu}_k(t)$.
  Observe $r_{k_t}(t)$.
  $B_{k_t} = B_{k_t} + c(t)c(t)^\top$, $f_{k_t} = f_{k_t} + c(t) r_{k_t}(t)$, $\hat{\mu}_{k_t} = B_{k_t}^{-1} f_{k_t}$.
end

reward $r_k(t)$ for choosing arm k at time t follows a parametric likelihood function $Pr(r(t)|\tilde{\mu})$. Following Agrawal and Goyal [AG13], the posterior distribution at time $t+1$, $Pr(\tilde{\mu}|r(t)) \propto Pr(r(t)|\tilde{\mu}) Pr(\tilde{\mu})$ is given by a multivariate Gaussian distribution $\mathcal{N}(\hat{\mu_k}(t+1), v^2 B_k(t+1)^{-1})$, where $B_k(t) = I_d + \sum_{\tau=1}^{t-1} c(\tau)c(\tau)^\top$, d is the size of the context vectors c, $v = R\sqrt{\frac{24}{z} d \cdot ln(\frac{1}{\gamma})}$ and we have $R > 0$, $z \in [0,1]$, $\gamma \in [0,1]$ constants, and $\hat{\mu}(t) = B_k(t)^{-1}(\sum_{\tau=1}^{t-1} c(\tau) r_k(\tau))$.

Every step t consists of generating a d-dimensional sample $\tilde{\mu_k}(t)$ from $\mathcal{N}(\hat{\mu_k}(t), v^2 B_k(t)^{-1})$ for each arm. We then decide which arm k to pull by solving for $\text{argmax}_{k \in [K]}\, c(t)^\top \tilde{\mu_k}(t)$. This means that at each time step we are selecting the arm that we expect to maximize the observed reward given a sample of our current beliefs over the distribution of rewards, $c(t)^\top \tilde{\mu_k}(t)$. We then observe the actual reward of pulling arm k, $r_k(t)$ and update our beliefs.

8.3.4 Problem Setting

In our setting, the agent is in multi-objective Markov decision processes (MOMDPs). Instead of the usual scalar reward function $R(s, a, s')$, a reward vector $\vec{R}(s, a, s')$ is present. The vector $\vec{R}(s, a, s')$ consists of l dimensions or components representing the different objectives, i.e., $\vec{R}(s, a, s') = (R_1(s, a, s'), \ldots, R_l(s, a, s'))$. However, not all components of the reward vector are observed in our setting. There is an objective $v \in [l]$ that is hidden, and the agent is only allowed to observe expert demonstrations to learn this objective.

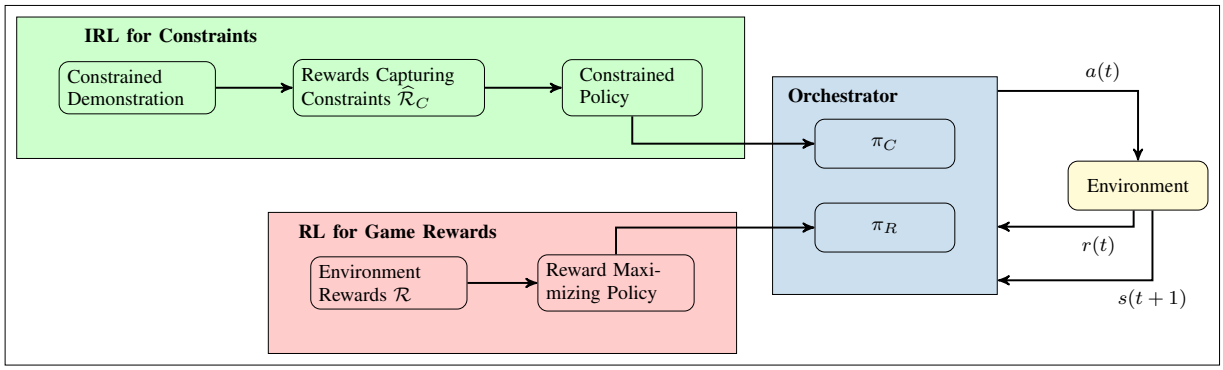


Figure 8.1: Overview of our system. At each time step the Orchestrator selects between two policies, π_C and π_R depending on the observations from the Environment. The two policies are learned before engaging with the environment. π_C is obtained using IRL on the demonstrations to learn a reward function that captures demonstrated constraints. The second, π_R is obtained by the agent through RL on the environment.

These demonstrations are given in the form of trajectories $\{Tr^{(1)}, Tr^{(2)}, \ldots, Tr^{(m)}\}$. To summarize, for some objectives, the agent has rewards observed from interaction with the environment, and for some objectives the agent has only expert demonstrations. The aim is still the same as single objective reinforcement learning, which is trying to maximize $\sum_{t=0}^{\infty} \gamma^t R_i(s_t, a_t, s_{t+1})$ for each $i \in [l]$.

8.4 Proposed Approach

The overall approach we propose, aggregation at the policy phase, is depicted by Figure 8.1. It has three main components. The first is the IRL component to learn the desired constraints (depicted in green in Figure 8.1). We apply IRL to the demonstrations depicting desirable behavior, to learn the underlying constraint rewards being optimized by the demonstrations. We then apply RL on these learned rewards to learn a strongly constraint-satisfying policy π_C. Next, we augment this with a pure reinforcement learning component (depicted in red in Figure 8.1). For this, we directly apply reinforcement learning to the original environment rewards to learn a domain reward maximizing policy π_R.

Now we have two policies: the constraint-obeying policy π_C and the reward-maximizing policy π_R. To combine these two, we use the third component, the orchestrator (depicted in blue in Figure 8.1). This is a contextual bandit algorithm that orchestrates the two policies, picking one of them to play at each point of time. The context is the state of the environment; the bandit decides which arm (policy) to play at each step. We use a modified CTS algorithm to train the bandit. The context of the bandit is given by features of the current state (for which we want to decide which policy to choose), i.e., $c(t) = \Upsilon(s_t) \in \mathbb{R}^d$.

The exact algorithm used to train the orchestrator is given in Algorithm 8.2. Apart from the fact that arms are policies (instead of atomic actions), the main difference from the CTS algorithm is the way rewards are fed into the bandit. For simplicity, we call the constraint policy π_C as arm 0 and the reward policy π_R as arm 1. We now go over

Algorithm 8.2. First, all the parameters are initialized as in the CTS algorithm (Line 1). For each time-step in the training phase (Line 3), we do the following. Pick an arm k_t according to the Thompson Sampling algorithm and the context $\Upsilon(s_t)$ (Lines 4 and 5). Play the action according to the chosen policy π_{k_t} (Line 6). This takes us to the next state s_{t+1}. We also observe two rewards (Line 7): (i) the original reward in environment, $r^R_{a_t}(t) = \mathcal{R}(s_t, a_t, s_{t+1})$ and (ii) the constraint rewards according to the rewards learnt by inverse reinforcement learning, i.e., $r^C_{a_t}(t) = \widehat{\mathcal{R}}_C(s_t, a_t, s_{t+1})$. $r^C_{a_t}(t)$ can intuitively be seen as the predicted reward (or penalty) for any constraint satisfaction (or violation) in this step.

Algorithm 8.2: Orchestrator Based Algorithm

1 **Initialize:** $B_k = I_d$, $\hat{\mu}_k = 0_d$, $f_k = 0_d$ for $k \in \{0, 1\}$.
2 **Observe** start state s_0.
3 **for** $t = 0, 1, 2, \ldots, (T-1)$ **do**
4 Sample $\tilde{\mu}_k(t)$ from $N(\hat{\mu}_k, v^2 B_k^{-1})$.
5 Pick arm $k_t = \arg\max_{k \in \{0,1\}} \Upsilon(s_t)^\top \tilde{\mu}_k(t)$.
6 Play corresponding action $a_t = \pi_{k_t}(s_t)$.
7 Observe rewards $r^C_{a_t}(t)$ and $r^R_{a_t}(t)$, and the next state s_{t+1}.
8 Define $r_{k_t}(t) = \lambda \left(r^C_{a_t}(t) + \gamma V^C(s_{t+1})\right) + (1-\lambda)\left(r^R_{a_t}(t) + \gamma V^R(s_{t+1})\right)$.
9 Update $B_{k_t} = B_{k_t} + \Upsilon(s_t)\Upsilon(s_t)^\top$, $f_{k_t} = f_{k_t} + \Upsilon(s_t) r_{k_t}(t)$, $\hat{\mu}_{k_t} = B_{k_t}^{-1} f_{k_t}$.
10 **end**

To train the contextual bandit to choose arms that perform well on both metrics (environment rewards and constraints), we feed it a reward that is a linear combination of $r^R_{a_t}(t)$ and $r^C_{a_t}(t)$ (Line 8). Another important point to note is that $r^R_{a_t}(t)$ and $r^C_{a_t}(t)$ are immediate rewards achieved on taking action a_t from s_t, they do not capture long term effects of this action. In particular, it is important to also look at the "value" of the next state s_{t+1} reached, since we are in the sequential decision making setting. Precisely for this reason, we also incorporate the value-function of the next state s_{t+1} according to both the reward maximizing component and constraint component (which encapsulate the long-term rewards and constraint satisfaction possible from s_{t+1}). This gives exactly Line 8, where V^C is the value-function according the constraint policy π_C, and V^R is the value-function according to the reward maximizing policy π_R.

In this equation, λ is a hyperparameter chosen by a user to decide how much to trade off environment rewards for constraint satisfaction. For example, when λ is set to 0, the orchestrator would always play the reward policy π_R, while for $\lambda = 1$, the orchestrator would always play the constraint policy π_C. For any value of λ in-between, the orchestrator is expected to pick policies at each point of time that would perform well on both metrics (weighed according to λ). Finally, for the desired reward $r_{k_t}(t)$ and the context $\Upsilon(s_t)$, the parameters of the bandit are updated according to the CTS algorithm (Line 9).

8.4.1 Alternative Approaches

Observe that in the proposed approach, we combine or "aggregate" the two objectives at the highest level, i.e., at the policy stage. Alternative approaches could be to combine the two objectives at lower levels, i.e., the reward stage or the demonstrations stage itself.

- **Aggregation at reward phase.** As before, we can perform inverse reinforcement learning to learn the underlying rewards capturing the desired constraints. Now, instead of learning a policy for each of the two reward functions (environment rewards and constraint rewards) followed by aggregating them, we could just combine the reward functions themselves. And then, we could learn a policy on these "aggregated" rewards that performs well on both the objectives, environment reward, and constraints. This process captures the intuitive idea of "incorporating the constraints into the environment rewards." Hence, if we were explicitly given the penalty of violating constraints this would be ideal. However, note that this is a *top-down* approach and in this study we want to focus on the example driven, or *bottoms-up* approach.
- **Aggregation at data phase.** Moving another step backward, we could aggregate the two objectives of play at the data phase. This could be performed as follows. We perform pure reinforcement learning as in the proposed approach given in Figure 8.1 (depicted in red). Once we have our reward maximizing policy π_R, we use it to generate numerous reward-maximizing demonstrations. Then, we combine these environment reward trajectories with the original constrained demonstrations, aggregating the two objectives in the process. And once we have the combined data, we can perform inverse reinforcement learning to learn the appropriate rewards, followed by reinforcement learning to learn the corresponding policy.

Aggregation at the policy phase is the proposed approach in the main paper, where we go all the way to the end of the pipeline, learning a policy for each of the objectives followed by aggregation. A similar parameter to λ there can be used by the reward aggregation and data aggregation approaches as well, to decide how to weigh the two objectives while performing the corresponding aggregation.

The question now is, "which of these aggregation procedures is the most useful?". The reason we use aggregation at the policy stage is to gain *interpretability*. Using an orchestrator to pick a policy at each point of time helps us identify which policy is being played at each point of time and also the reason for which it is being chosen (in the case of an interpretable orchestrator, which it is in our case).

8.5 Demonstration on Pac-Man

We demonstrate the applicability of the proposed algorithm using the classic game of Pac-Man.

8.5.1 Details of the Domain

The layout of Pac-Man we use is given in Figure 8.2. The rules for the environment (adopted

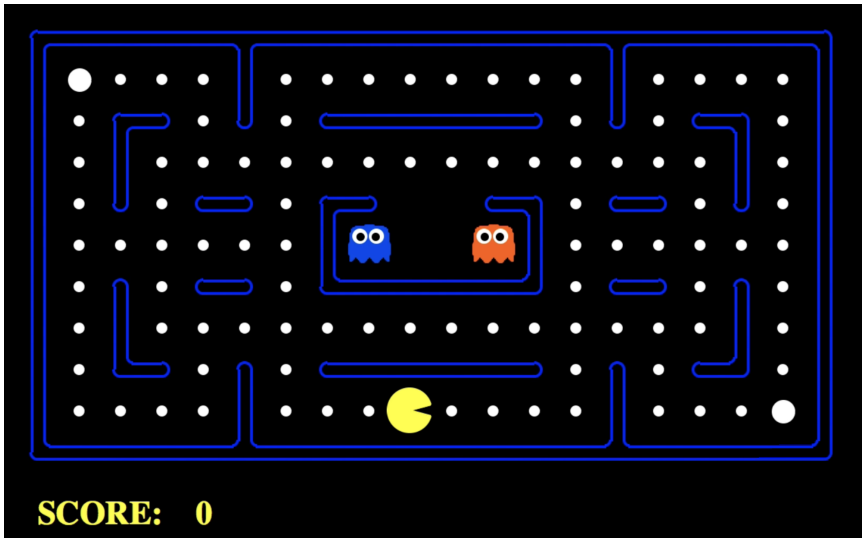


Figure 8.2: Layout of Pac-Man

from Berkeley AI Pac-Man[2]) are as follows. The goal of the agent is to eat all the dots in the maze, known as Pac-Dots, as soon as possible while simultaneously avoiding collision with ghosts. On eating a Pac-Dot, the agent obtains a reward of +10. On successfully eating all the Pac-Dots, the agent obtains a reward of +500. In the meantime, the ghosts roam the maze trying to kill Pac-Man. On collision with a ghost, Pac-Man loses the game and gets a reward of −500. The game also has two special dots called Power Pellets in the corners of the maze, which on consumption, give Pac-Man the temporary ability of "eating" ghosts. During this phase, the ghosts are in a "scared" state for 40 frames and move at half their speed. On eating a ghost, the agent gets a reward of +200, the ghost returns to the center box and returns to its normal "unscared" state. Finally, there is a constant time-penalty of −1 for every step taken.

For the sake of demonstration of our approach, we define *not eating ghosts* as the desirable constraint in the game of Pac-Man. However, recall that this constraint is not given explicitly to the agent, but only through examples. To play optimally in the original game one should eat ghosts to earn bonus points, but doing so is being demonstrated as undesirable. Hence, the agent has to combine the goal of collecting the most points while not eating ghosts.

8.5.2 Details of the Pure RL

For the reinforcement learning component, we use Q-learning with linear function approximation as described in Section 8.3.1. Some of the features we use for an $\langle s, a \rangle$ pair (for the $\boldsymbol{\psi}(s, a)$ function) are: "whether food will be eaten", "distance of the next closest food", "whether a scared (unscared) ghost collision is possible" and "distance of the closest scared (unscared) ghost".

[2] http://ai.berkeley.edu/project_overview.html

For the layout of Pac-Man we use (shown in Figure 8.2), an upper bound on the maximum score achievable in the game is 2170. This is because there are 97 Pac-Dots, each ghost can be eaten at most twice (because of two capsules in the layout), Pac-Man can win the game only once and it would require more than 100 steps in the environment. On playing a total of 100 games, our reinforcement learning algorithm (the reward maximizing policy π_R) achieves an average game score of 1675.86, and the maximum score achieved is 2144. We mention this here, so that the results in Section 8.6 can be seen in appropriate light.

8.5.3 Details of the IRL

For inverse reinforcement learning, we use the linear IRL algorithm as described in Section 8.3.2. For Pac-Man, observe that the original reward function $\mathcal{R}(s, a, s')$ depends only on the following factors: "number of Pac-Dots eating in this step (s, a, s')", "whether Pac-Man has won in this step", "number of ghosts eaten in this step" and "whether Pac-Man has lost in this step". For our IRL algorithm, we use exactly these as the features $\boldsymbol{\phi}(s, a, s')$. As a sanity check, when IRL is run on environment reward optimal trajectories (generated from our policy π_R), we recover something very similar to the original reward function $\mathcal{R}$. In particular, the weights of the reward features learned is given by $1/1000[+2.44, +138.80, +282.49, -949.17]$, which when scaled is almost equivalent to the true weights $[+10, +500, +200, -500]$ in terms of their optimal policies. The number of trajectories used for this is 100.

Ideally, we would prefer to have the constrained demonstrations given to us by humans, but for the sake of simplicity we generate them synthetically as follows. We learn a policy $\pi_C^\star$ by training it on the game with the original reward function $\mathcal{R}$ augmented with a very high negative reward (-1000) for eating ghosts. This causes $\pi_C^\star$ to play well in the game while avoiding eating ghosts as much as possible.[3] Now, to emulate erroneous human behavior, we use $\pi_C^\star$ with an error probability of 3%. That is, at every time step, with 3% probability we pick a completely random action, and otherwise follow $\pi_C^\star$. This gives us our constrained demonstrations, on which we perform inverse reinforcement learning to learn the rewards capturing the constraints. The weights of the reward function learned is given by $1/1000[+2.84, +55.07, -970.59, -234.34]$, and it is evident that it has learned that eating ghosts strongly violates the favorable constraints. The number of demonstrations used for this is 100. We scale these weights to have a similar L_1 norm as the original reward weights $[+10, +500, +200, -500]$, and denote the corresponding reward function by $\widehat{\mathcal{R}}_C$.

Finally, running reinforcement learning on these rewards $\widehat{\mathcal{R}}_C$, gives us our constraint policy π_C. On playing a total of 100 games, π_C achieves an average game score of 1268.52 and eats just 0.03 ghosts on an average. Note that, when eating ghosts is prohibited in the domain, an upper bound on the maximum score achievable is 1370.

[3] We do this only for generating demonstrations. In real domains, we would not have access to the exact constraints that we want to be satisfied, and hence a policy like $\pi_C^\star$ cannot be learned; learning from human demonstrations would then be essential.

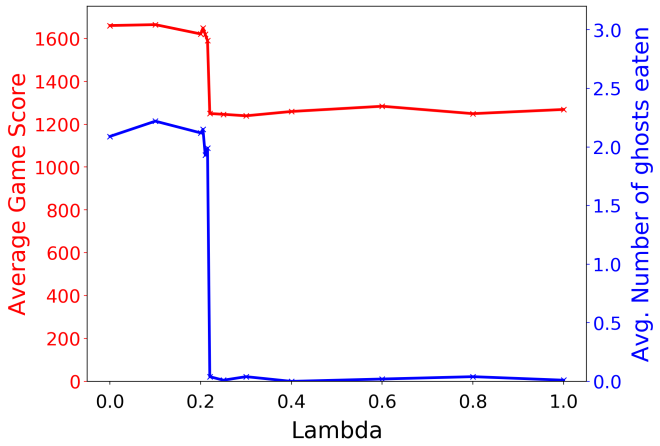


Figure 8.3: Both performance metrics as λ is varied. The red curve depicts the average game score achieved, and the blue curve depicts the average number of ghosts eaten.

8.5.4 Details of the Contextual Bandit

The features of the state we use for context $c(t)$ are: (i) A constant 1 to represent the bias term, and (ii) The distance of Pac-Man from the closest scared ghost in s_t. One could use a more sophistical context with many more features, but we use this restricted context to demonstrate a very interesting behavior (shown in Section 8.6).

8.6 Evaluation

We measure performance on two metrics, (i) the total score achieved in the game (the environment rewards) and (ii) the number of ghosts eaten (the constraint violation). We also observe how these metrics vary with λ. For each value of λ, the orchestrator is trained for 100 games. The results are shown in Figure 8.3. Each point in the graph is averaged over 100 test games.

The graph shows a very interesting pattern. When λ is at most than 0.215, the agent eats a lot of ghosts, but when it is above 0.22, it eats almost no ghosts. In other words, there is a value λ_o which behaves as a tipping point, across which there is drastic change in behavior. Beyond the threshold, the agent learns that eating ghosts is not worth the score it is getting and so it avoids eating as much as possible. On the other hand, when λ is smaller than λ_o, it learns the reverse and eats as many ghosts as possible.

Policy-switching. As mentioned before, one important property of our approach is interpretability, we know exactly which policy is being played at each time. For moderate values of $\lambda > \lambda_o$, the orchestrator learns a very interesting policy-switching technique: whenever at least one of the ghosts in the domain is scared, it plays π_C, but if no ghosts are scared, it plays π_R. In other words, it starts the game playing π_R until a capsule is eaten. As soon as the first capsule is eaten, it switches to π_C until the scared timer runs off. Then it switches back to π_R until another capsule is eaten, and so on. It has learned a very intuitive behavior: when there is no scared ghost, there is no possibility of violating

constraints. Hence, the agent is as greedy as possible (i.e., play π_R). However, when there are scared ghosts, it is better to be safe (i.e., play π_C).

8.7 Discussion and Extensions

In this paper, we have considered the problem of autonomous agents learning policies that are constrained by implicitly-specified norms and values while still optimizing their policies with respect to environmental rewards. We have taken an approach that combines IRL to determine constraint-satisfying policies from demonstrations, RL to determine reward-maximizing policies, and a contextual bandit to orchestrate between these policies in a transparent way. This proposed architecture and approach for the problem is novel. It also requires a novel technical contribution in the contextual bandit algorithm because the arms are policies rather than atomic actions, thereby requiring rewards to account for sequential decision making. We have demonstrated the algorithm on the Pac-Man video game and found it to perform interesting switching behavior among policies.

We feel that the contribution herein is only the starting point for research in this direction. We have identified several avenues for future research, especially with regards to IRL. We can pursue deep IRL to learn constraints without hand-crafted features, develop an IRL that is robust to noise in the demonstrations, and research IRL algorithms to learn from just one or two demonstrations (perhaps in concert with knowledge and reasoning). In real-world settings, demonstrations will likely be given by different users with different versions of abiding behavior; we would like to exploit the partition of the set of traces by user to improve the policy or policies learned via IRL. Additionally, the current orchestrator selects a single policy at each time, but more sophisticated policy aggregation techniques for combining or mixing policies is possible. Lastly, it would be interesting to investigate whether the policy aggregation rule (λ in the current proposal) can be learned from demonstrations.

Chapter 9

Inverse Reinforcement Learning From Like-Minded Teachers

We study the problem of learning a policy in a Markov decision process (MDP) based on observations of the actions taken by multiple teachers. We assume that the teachers are like-minded in that their reward functions — while different from each other — are random perturbations of an underlying reward function. Under this assumption, we demonstrate that inverse reinforcement learning algorithms that satisfy a certain property — that of *matching feature expectations* — yield policies that are approximately optimal with respect to the underlying reward function, and that no algorithm can do better in the worst case. We also show how to efficiently recover the optimal policy when the MDP has one state — a setting that is akin to multi-armed bandits. Finally, we support this with experiments on non-trivial bandit problems, with varying parameters.

9.1 Introduction

A *Markov decision process (MDP)* is a formal specification of a sequential decision making environment, which consists of a set of states, a set of actions, a reward function, and a stochastic transition function. *Reinforcement learning (RL)* deals with learning a *policy* in an MDP — which specifies a possibly randomized action that is taken in each state — to maximize cumulative reward.

RL has long history in AI [SB98a; KLM96], as well as in many other disciplines. But in recent years, interest in the area has exploded, in part due to breakthroughs in game playing [Mni+15; Sil+16] and fast-growing applications to robotics [KBP13]. It is safe to say that, nowadays, RL is widely considered to be one of the basic building blocks in the construction of intelligent agents.

While most work in the area focuses on maximizing a given reward function, some settings require the AI system to emulate the behavior of an expert or teacher [NR00a; AN04a] — this is known as *inverse reinforcement learning (IRL)*. The idea is to observe an agent executing a policy in an MDP, where everything is known to the learner except

the reward function, and extract a reward function that is most likely to be the one being optimized by the agent. Using this reward function — and knowledge of the other components of the MDP — the agent can easily compute an optimal policy to follow.

Our point of departure is that we are interested in IRL from multiple agents rather than a single agent. Specifically, we observe n different agents executing policies that are optimal for their individual reward functions. Our approach is to aggregate these observations into a single policy, by applying an inverse reinforcement learning algorithm to the set of all observations.

However, if individual agents have wildly divergent reward functions then the aggregate policy may not represent coherent behavior. In addition, to formally reason about the quality of the optimal policy, we need to relate it to some notion of ground truth. For these reasons, we assume that the agents are *like-minded*, in that individual reward functions are nothing but noisy versions of an underlying reward function.

In summary, our research challenge is this: *Given observations from policies that are optimal with respect to different reward functions, each of which is a perturbation of an underlying reward function, identify IRL algorithms that can recover a good policy with respect to the underlying reward function.*

We believe that this problem is both natural and general. To further motivate it, though, let us briefly instantiate it in the context of beneficial AI. One of the prominent approaches in this area is to align the values of the AI system with the values of a human through IRL [RDT15; Had+16]. Our extension to multiple agents would allow the alignment of the system with the values of *society*.

A compelling aspect of this instantiation is that, if we think of the underlying reward function as embodying a common set of moral propositions, then our technical assumption of like-minded agents can be justified through the *linguistic analogy*, originally introduced by Rawls [Raw71]. It draws on the work of Chomsky [Cho65], who argued that competent speakers have a set of grammatical principles in mind, but their linguistic behavior is hampered by "grammatically irrelevant conditions such as memory limitations, distractions, shifts of attention and interest, and errors." Analogously, Rawls claimed, humans have moral rules — a common "moral grammar" — in our minds, but, due to various limitations, our moral behavior is only an approximation thereof. Interestingly, this theory lends itself to empirical experimentation, and, indeed, it has been validated through work in moral psychology [Mik11].

Our Model and Results. We start from a common IRL setup: each reward function is associated with a weight vector $\mathbf{w}$, such that the reward for taking a given action in a given state is the dot product of the weight vector and the feature vector of that state-action pair. The twist is that there is an underlying reward function represented by a weight vector $\mathbf{w}^\star$, and each of the agents is associated with a weight vector $\mathbf{w}_i$, which induces an optimal policy π_i. We observe a trajectory from each π_i.

In Section 9.3, we focus on competing with a uniform mixture over the optimal policies of the agents, $\pi_1, \ldots, \pi_n$ (for reasons that we explicate momentarily). We can do this because the observed trajectories are "similar" to the uniform mixture, in the sense that their feature vectors — the discounted frequencies of the features associated with the observed state-action pairs — are close to that of the uniform mixture policy. Therefore, due to

the linearity of the reward function, any policy whose feature expectations approximately match those of the observed trajectories must be close to the uniform mixture with respect to $\mathbf{w}^\star$. We formalize this idea in Theorem 9.3.2, which gives a lower bound on the number of agents and length of observed trajectories such that any policy that $\epsilon/3$-matches feature expectations is ϵ-close to the uniform mixture. Furthermore, we identify two well-known IRL algorithms, Apprenticeship Learning [AN04a] and Max Entropy [Zie+08], which indeed output policies that match the feature expectations of the observed trajectories, and therefore enjoy the guarantees provided by this theorem.

Needless to say, competing with the uniform mixture is only useful insofar as this benchmark exhibits "good" performance. We show that this is indeed the case in Section 9.4, assuming (as stated earlier) that each weight vector $\mathbf{w}_i$ is a noisy perturbation of $\mathbf{w}^\star$. Specifically, we first establish that, under relatively weak assumptions on the noise, it is possible to bound the difference between the reward of the uniform mixture and that of the optimal policy (Theorem 9.4.1). More surprisingly, Theorem 9.4.3 asserts that in the worst case it is impossible to outperform the uniform mixture, by constructing an MDP where the optimal policy cannot be identified — even if we had an infinite number of agents and infinitely long trajectories! Putting all of these results together, we conclude that directly running an IRL algorithm that matches feature expectations on the observed trajectories is a sensible approach to our problem.

Nevertheless, it is natural to ask whether it is possible to outperform the uniform mixture in typical instances. In Section 9.5 we show that this is indeed the case; in fact, we are able to recover the optimal policy whenever it is identifiable, albeit under stringent assumptions — most importantly, that the MDP has only one state. This leads to challenge that we call the *inverse multi-armed bandit problem*. To the best of our knowledge, this problem is novel; its study contributes to the (relatively limited) understanding of scenarios where it is possible to outperform teacher demonstrations.

Related work. The most closely related work deals with IRL when the observations come from an agent who acts according to multiple *intentions*, each associated with a different reward function [Bab+11; CK12]. The main challenge stems from the need to cluster the observations — the observations in each cluster are treated as originating from the same policy (or intention). By contrast, clustering is a nonissue in our framework. Moreover, our assumption that each $\mathbf{w}_i$ is a noisy perturbation of $\mathbf{w}^\star$ allows us to provide theoretical guarantees.

Further afield, there is a body of work on robust RL and IRL under reward uncertainty [GLD00; RB09; RB10], noisy rewards [ZLN14], and corrupted rewards [Eve+17]. Of these papers the closest to ours is that of Zheng et al. [ZLN14], who design robust IRL algorithms under *sparse* noise, in the sense that only a small fraction of the observations are anomalous; they do not provide theoretical guarantees. Our setting is quite different, as very few observations would typically be associated with a near-perfect policy.

9.2 MDP Terminology

We assume the environment is modeled as an MDP $\{S, A, T, \gamma, D\}$ with an unknown reward function. S is a finite set of states; A is a finite set of actions; $T(s, a, s')$ is the state transition probability of reaching state s' from state s when action a is taken; $\gamma \in [0, 1)$ is the discount factor; and D the initial-state distribution, from which the start state s_0 is drawn for every trajectory.

As is standard in the literature [AN04a], we assume that there is a function $\phi : S \times A \to \mathbb{R}^d$ that maps state-action pairs to their real-valued features. We also overload notation, and say that the feature vector of a trajectory $\tau = \{(s_0, a_0), (s_1, a_1), \ldots, (s_L, a_L)\}$ is defined as $\phi(\tau) = \sum_{t=0}^{L} \gamma^t \phi(s_t, a_t)$.

We make the standard assumption that the immediate reward of executing action a from state s is linear in the features of the state-action pair, i.e. $r^{\mathbf{w}}(s, a) = \mathbf{w}^\intercal \phi(s, a)$. This has a natural interpretation: ϕ represents the different factors, and $\mathbf{w}$ weighs them in varying degrees.

Let μ denote the feature expectation of policy π, that is, $\mu(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t)|\pi]$, where π defines the action a_t taken from state s_t, and the expectation is taken over the transition probabilities $T(s_t, a_t, s_{t+1})$. Hence, the cumulative reward of a policy π under weight $\mathbf{w}$ can be rewritten as:

$$R^{\mathbf{w}}(\pi) = \mathbb{E}_{s_0 \sim D}[V^{\pi}(s_0)] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r^{\mathbf{w}}(s_t, a_t) \middle| \pi\right] = \mathbf{w}^\intercal \cdot \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a) \middle| \pi\right] = \mathbf{w}^\intercal \mu(\pi).$$

Let $P_\pi(s, t)$ denote the probability of getting to state s at time t under policy π. Then, the cumulative reward $R^{\mathbf{w}}$ is

$$R^{\mathbf{w}}(\pi) = \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} P_\pi(s, t) r^{\mathbf{w}}(s, \pi(s)).$$

9.3 Approximating the Uniform Mixture

We consider an environment with n agents $N = \{1, \ldots, n\}$. Furthermore, the reward function of each agent $i \in N$ is associated with a weight vector $\mathbf{w}_i$, and, therefore, with a reward function $r^{\mathbf{w}_i}$. This determines the optimal policy π_i executed by agent i, from which we observe the trajectory τ_i, which consists of L steps. We observe such a trajectory for each $i \in N$, giving us trajectories $\{\tau_1, ..., \tau_n\}$.

As we discussed in Section 9.1, we assume that the reward function associated with each agent is a noisy version of an underlying reward function. Specifically, we assume that there exists a ground truth weight vector $\mathbf{w}^\star$, and for each agent $i \in N$ we let $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, where $\boldsymbol{\eta}_i$ is the corresponding noise vector; we assume throughout that $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_n$ are i.i.d. Following Abbeel and Ng [AN04a], we also assume in some of our results (when stated explicitly) that $\|\mathbf{w}^\star\|_2 \leq 1$ and $\|\phi(s, a)\|_\infty \leq 1$.

Let us denote by π^u the *uniform mixture* over the policies $\pi_1, \ldots, \pi_n$, that is, the (randomized) policy that, in each trajectory, selects one of these policies uniformly at random and executes it throughout the trajectory.

Our goal in this section is to "approximate" the uniform mixture (and we will justify this choice in subsequent sections). To do so, we focus on IRL algorithms that "match feature expectations." Informally, the property of interest is that the feature expectations of the policy match the (discounted) feature vectors of observed trajectories. This idea is already present in the IRL literature, but it is helpful to define it formally, as it allows us to identify specific IRL algorithms that work well in our setting.

Definition 9.3.1. Given n trajectories $\tau_1, ..., \tau_n$, a (possibly randomized) policy π ϵ-*matches their feature expectations* if and only if $\|\mu(\pi) - \frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\|_2 \leq \epsilon$.

In a nutshell, due to the linearity of the reward function, two policies that have the same feature expectations have the same reward. Therefore, if the observed trajectories closely mimic the feature expectations of π_u, and a policy $\tilde{\pi}$ matches the feature expectations of the observed trajectories, then the reward of $\tilde{\pi}$ would be almost identical to that of π^u. This is formalized in the following theorem, whose proof is relegated to Appendix G.2.

Theorem 9.3.2. *Assume that* $\|\phi(s,a)\|_\infty \leq 1$ *for all* $s \in S, a \in A$. *Let* $\mathbf{w}^\star$ *such that* $\|\mathbf{w}^\star\|_2 \leq 1$, *fix any* $\mathbf{w}_1, \ldots, \mathbf{w}_n$, *and, for all* $i \in N$, *let* τ_i *be a trajectory of length* L *sampled by executing* π_i. *Let* $\tilde{\pi}$ *be a policy that* $\epsilon/3-$*matches the feature expectation of these trajectories. If*

$$n \geq \frac{72 \ln\left(\frac{2}{\delta}\right) d}{\epsilon^2 (1-\gamma)^2} \quad \text{and} \quad L \geq \log_{1/\gamma} \frac{3\sqrt{d}}{(1-\gamma)\epsilon}$$

then, with probability at least $1-\delta$, *it holds that* $\left|R^{\mathbf{w}^\star}(\tilde{\pi}) - R^{\mathbf{w}^\star}(\pi^u)\right| \leq \epsilon$.

Note that the required number of agents n may be significant; fortunately, we can expect access to data from many agents in applications of interest. For example, Noothigattu et al. [Noo+18] built a system that decides ethical dilemmas based on data collected from 1.3 million people.

To apply Theorem 9.3.2, we need to use IRL algorithms that match feature expectations. We have identified two algorithms that satisfy this property: the *Apprenticeship Learning* algorithm of Abbeel and Ng [AN04a], and the *Max Entropy* algorithm of Ziebart et al. [Zie+08]. For completeness we present these algorithms, and formally state their feature-matching guarantees, in Appendix G.1.

9.4 How Good is the Uniform Mixture?

In Section 9.3 we showed that it is possible to (essentially) match the performance of the uniform mixture with respect to the ground truth reward function. In this section we justify the idea of competing with the uniform mixture in two ways: first, we show that the uniform mixture approximates the optimal policy under certain assumptions on the noise, and, second, we prove that in the worst case it is actually impossible to outperform the uniform mixture.

9.4.1 The Uniform Mixture Approximates the Optimal Policy

Recall that for all $i \in n$, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$. It is clear that without imposing some structure on the noise vectors $\boldsymbol{\eta}_i$, no algorithm would be able to recover a policy that does well with respect to $\mathbf{w}^\star$.

Let us assume, then, that the noise vectors $\boldsymbol{\eta}_i$ are such that the η_{ik} are independent and each η_{ik}^2 is sub-exponential. Formally, a random variable X with mean $u = \mathbb{E}[X]$ is *sub-exponential* if there are non-negative parameters (ν, b) such that $\mathbb{E}\left[\exp\left(\lambda(X-u)\right)\right] \leq \exp\left(\nu^2\lambda^2/2\right)$ for all $|\lambda| < 1/b$. This flexible definition simply means that the moment generating function of the random variable X is bounded by that of a Gaussian in a neighborhood of 0. Note that if a random variable is sub-Gaussian, then its square is sub-exponential. Hence, our assumption is strictly weaker than assuming that each η_{ik} is sub-Gaussian.

Despite our assumption about the noise, it is *a priori* unclear that the uniform mixture would do well. The challenge is that the noise operates on the coordinates of the individual weight vectors, which in turn determine individual rewards, but, at first glance, it seems plausible that relatively small perturbations of rewards would lead to severely suboptimal policies. Our result shows that this is not the case: π^u is approximately optimal with respect to $R^{\mathbf{w}^\star}$, in expectation.

Theorem 9.4.1. *Assume that $\|\phi(s,a)\|_\infty \leq 1$ for all $s \in S, a \in A$. Let $\mathbf{w}^\star$ such that $\|\mathbf{w}^\star\|_2 \leq 1$, and suppose that $\mathbf{w}_1, ..., \mathbf{w}_n$ are drawn from i.i.d. noise around $\mathbf{w}^\star$, i.e., $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, where each of its coordinates is such that η_{ik}^2 is an independent sub-exponential random variable with parameters (ν, b). Then*

$$\mathbb{E}[R^{\mathbf{w}^\star}(\pi^u)] \geq R^{\mathbf{w}^\star}(\pi^\star) - O\left(d\sqrt{u} + \nu\sqrt{\frac{d}{u}} + \frac{b}{\sqrt{u}}\right),$$

where $u = \frac{1}{d}\sum_{k=1}^{d}\mathbb{E}\left[\eta_{ik}^2\right]$, and the expectation is taken over the noise.

The exact expression defining the gap between $\mathbb{E}[R^{\mathbf{w}^\star}(\pi^u)]$ and $R^{\mathbf{w}^\star}(\pi^\star)$ can be found in the proof of Theorem 9.4.1, which appears in Appendix G.3; we give the asymptotic expression in the theorem's statement because it is easier to interpret. As one might expect, this gap increases as ν or b is increased (and, in a linear fashion). This is intuitive because a smaller ν or b imposes a strictly stronger assumption on the sub-exponential random variable (and its tails).

To gain more insight, we analyze the upper bound on the gap when η_{ik} follows a Gaussian distribution, that is, $\eta_{ik} \sim \mathcal{N}(0, \sigma^2)$. Note that this implies that η_{ik}^2 follows a χ_1^2 distribution scaled by σ^2; a χ_1^2 distributed random variable is known to be sub-exponential with parameters $(2, 4)$, and hence this implies that η_{ik}^2 is sub-exponential with parameters $(2\sigma^2, 4\sigma^2)$. Further, in this case, $u = \mathbb{E}\left[\eta_{ik}^2\right] = \sigma^2$. Plugging these quantities into the upper bound of Theorem 9.4.1 shows that the gap is bounded by $O(d\sigma)$.

Theorem 9.4.1 shows that the gap depends linearly on the number of features d. An example given in Appendix G.4 shows that this upper bound is tight. Nevertheless, the tightness holds in the worst case, and one would expect the practical performance of the uniform mixture to be very good. To corroborate this intuition, we provide (unsurprising) experimental results in Appendix G.5.

9.4.2 It is Impossible to Outperform the Uniform Mixture in the Worst Case

An ostensible weakness of Theorem 9.4.1 is that even as the number of agents n goes to infinity, the reward of the uniform mixture may not approach that of the optimal policy, that is, there is a persistent gap. The example given in Section 9.4.1 shows the gap is not just an artifact of our analysis. This is expected, because the data contains some agents with suboptimal policies π_i, and a uniform mixture over these suboptimal policies must itself be suboptimal.

It is natural to ask, therefore, whether it is generally possible to achieve performance arbitrarily close to $\pi^\star$ (at least in the limit that n goes to infinity). The answer is negative. In fact, we show that — in the spirit of *minimax optimality* [HL50; PM02] — one cannot hope to perform better than π^u itself in the worst case. Intuitively, there exist scenarios where it is impossible to tell good and bad policies apart by looking at the data, which means that the algorithm's performance depends on what can be gleaned from the "average data".

This follows from a surprising[1] result that we think of as "non-identifiability" of the optimal policy. To describe this property, we introduce some more notation. The distribution over the weight vector of each agent i, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$, in turn induces a distribution over the optimal policy π_i executed by each agent. Denote this distribution by $\mathcal{P}(\mathbf{w}^\star)$.[2] Hence, each agent's optimal policy π_i is just a sample from this distribution $\mathcal{P}(\mathbf{w}^\star)$. In particular, as the number of agents goes to infinity, the empirical distribution of their optimal policies would exactly converge to $\mathcal{P}(\mathbf{w}^\star)$.

For the rest of this section, we make minimal assumptions on the noise vector $\boldsymbol{\eta}_i$. In particular, we merely assume that $\boldsymbol{\eta}_i$ follows a continuous distribution and that each of its coordinates is i.i.d. We are now ready to state our non-identifiability lemma.

Lemma 9.4.2 (non-identifiability)**.** *For every continuous distribution $\mathcal{D}$ over $\mathbb{R}$, if η_{ik} is independently sampled from $\mathcal{D}$ for all $i \in N$ and $k \in [d]$, then there exists an MDP and weight vectors $\mathbf{w}_a^\star$, $\mathbf{w}_b^\star$ with optimal policies $\pi_a^\star$, $\pi_b^\star$, respectively, such that $\pi_a^\star \neq \pi_b^\star$ but $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$.*

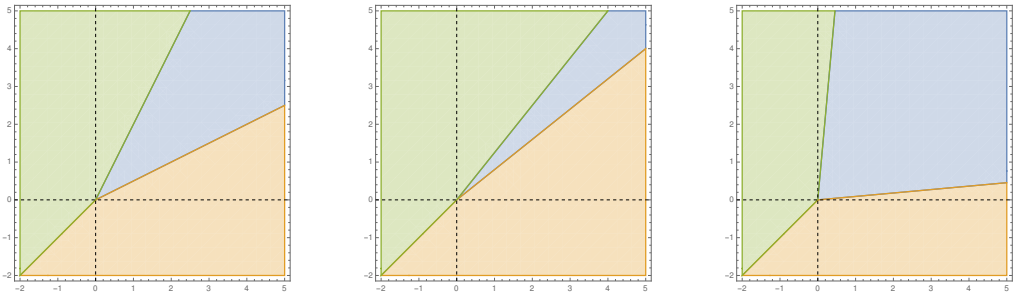
Even if we had an infinite number of trajectories in our data, and even if we knew the exact optimal policy played by each player i, this information would amount to knowing $\mathcal{P}(\mathbf{w}^\star)$. Hence, if there exist two weight vectors $\mathbf{w}_a^\star$, $\mathbf{w}_b^\star$ with optimal policies $\pi_a^\star$, $\pi_b^\star$ such that $\pi_a^\star \neq \pi_b^\star$ and $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$, then we would not be able to identify whether the optimal policy is $\pi_a^\star$ or $\pi_b^\star$ regardless of how much data we had.

The proof of Lemma 9.4.2 is relegated to Appendix G.6. Here we provide a proof sketch.

Proof sketch of Lemma 9.4.2. The intuition for the lemma comes from the construction of an MDP with three possible policies, all of which have probability 1/3 under $\mathcal{P}(\mathbf{w}^\star)$, even though one is better than the others. This MDP has a single state s, and three actions $\{a, b, c\}$ that lead back to s. Denote the corresponding policies by π_a, π_b, π_c. Let the feature

[1] At least it was surprising for us — we spent significant effort trying to prove the opposite result!

[2] Note that this distribution does not depend on i itself since the noise $\boldsymbol{\eta}_i$ is i.i.d. across the different agents.



$\delta = 1$ $\delta = 0.25$ $\delta = 10$

Figure 9.1: Regions of each optimal policy for different values of δ. Blue depicts the region where π_a is optimal, orange is where π_b is optimal, and green is where π_c is optimal.

expectations be $\phi(s,a) = [0.5, 0.5], \phi(s,b) = [1, -\delta/2], \phi(s,c) = [-\delta/2, 1]$, where $\delta > 0$ is a parameter. Let the ground truth weight vector be $\mathbf{w}^\star = (v_o, v_o)$, where v_o is such that the noised weight vector $\mathbf{w} = \mathbf{w}^\star + \boldsymbol{\eta}$ has probability strictly more than $1/3$ of lying in the first quadrant; such a value always exists for any noise distribution that is continuous and i.i.d. across coordinates.

Let us look at weight vectors $\mathbf{w}$ for which each of the three policies π_a, π_b and π_c are optimal. π_a is the optimal policy when $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_b$ and $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) > 0$ and $\mathbf{w}^\intercal(1+\delta, -1) > 0$. Similarly, we can reason about the regions where π_b and π_c are optimal. These regions are illustrated in Figure 9.1 for different values of δ. Informally, as δ is decreased, the lines separating (π_a, π_c) and (π_a, π_b) move closer to each other (as shown for $\delta = 0.25$), while as δ is increased, these lines move away from each other (as shown for $\delta = 10$). By continuity and symmetry, there exists δ such that the probability of each of the regions (with respect to the random noise) is exactly $1/3$, showing that the MDP has the desired property.

To complete the proof of the lemma, we extend the MDP by adding two more features to the existing two. By setting these new features appropriately (in particular, by cycling the two original features across the arms), we can show that the two weight vectors $\mathbf{w}_a^\star = (v_o, v_o, 0, 0)$ and $\mathbf{w}_b^\star = (0, 0, v_o, v_o)$ lead to $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) = \mathcal{P}(\mathbf{w}_b^\star)$, even though their corresponding optimal policies are π_a and π_b, respectively. □

For the next theorem, therefore, we can afford to be "generous:" we will give the algorithm (which is trying to compete with π^u) access to $\mathcal{P}(\mathbf{w}^\star)$, instead of restricting it to sampled trajectories. Formally, the theorem holds for any algorithm that takes a distribution over policies as input, and returns a randomized policy.

Theorem 9.4.3. *For every continuous distribution $\mathcal{D}$ over $\mathbb{R}$, if η_{ik} is independently sampled from $\mathcal{D}$ for all $i \in N$ and $k \in [d]$, then there exists an MDP such that for any algorithm $\mathcal{A}$ from distributions over policies to randomized policies, there exists a ground truth weight vector $\mathbf{w}^\star$ such that $R^{\mathbf{w}^\star}(\mathcal{A}(\mathcal{P}(\mathbf{w}^\star))) \leq R^{\mathbf{w}^\star}(\pi^u) < R^{\mathbf{w}^\star}(\pi^\star)$.*

In words, the constructed instance is such that, even given infinite data, no algorithm can outperform the uniform mixture, and, moreover, the reward of the uniform mixture is bounded away from the optimum. The theorem's proof is given in Appendix G.7.

9.5 An Algorithm for the Inverse Multi-Armed Bandit Problem

In Section 9.4, we have seen that it is impossible to outperform the uniform mixture in the worst case, as the optimal policy is not identifiable. However, it is natural to ask whether the optimal policy can be practically recovered when it is identifiable. In this section we give a positive answer, albeit in a restricted setting.

Specifically, we focus on the multi-armed bandit problem, which is an MDP with a single state. Note that the non-identifiability result of Lemma 9.4.2 still holds in this setting, as the example used in its proof is an MDP with a single state. Hence, even in this setting of bandits, it is impossible to outperform the uniform mixture in the worst case. However, we design an algorithm that can guarantee optimal performance when the problem is identifiable, under some additional conditions.

Like the more general setting of the previous sections, there exists a ground truth weight vector $\mathbf{w}^\star$, and for each agent $i \in N$, $\mathbf{w}_i = \mathbf{w}^\star + \boldsymbol{\eta}_i$. For this section, we assume the noise vector $\boldsymbol{\eta}_i$ to be Gaussian, and i.i.d. across agents as well as coordinates. In particular, $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2 I_d)$, and independent across i.

The bandit setting is equivalent to a single-state MDP, and hence the components S, T, γ and D are moot. Instead, there are m arms to pull, denoted by $A = \{1, 2, \ldots, m\}$. Similar to our original feature function ϕ, we now have features $\mathbf{x}_j \in \mathbb{R}^d$ associated with arm j, for each $j \in A$. Although in standard stochastic bandit problems we have a reward sampled from a distribution when we pull an arm, we care only about its mean reward in this section. For weight vector $\mathbf{w}$, the (mean) reward of pulling arm j is given by $r^{\mathbf{w}}(j) = \mathbf{w}^\intercal \mathbf{x}_j$. For each agent i (with weight vector $\mathbf{w}_i$), we assume that we observe the optimal arm being played by this agent, i.e., $\tilde{a}_i = \operatorname{argmax}_{j \in A} \mathbf{w}_i^\intercal \mathbf{x}_j$.

We observe the dataset $\mathcal{D} = \{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_n\}$ which is the set of optimal arms played by the agents. Define $\mathcal{Q}(\mathbf{w}^\star)$ to be the distribution over optimal arms induced when the ground truth weight vector is $\mathbf{w}^\star$. In particular, ground truth weight vector $\mathbf{w}^\star$ induces a distribution over the noised weight vector of each agent (via $\mathbf{w} = \mathbf{w}^\star + \eta$), which in turn induces a distribution over the optimal arm that would be played, which we call $\mathcal{Q}(\mathbf{w}^\star)$ — analogously to the $\mathcal{P}(\mathbf{w}^\star)$ of Section 9.4. Observe that the dataset $\mathcal{D}$ could be rewritten as a distribution over arms, $\tilde{\mathcal{Q}} = (\tilde{\mathcal{Q}}_1, \tilde{\mathcal{Q}}_2, \ldots, \tilde{\mathcal{Q}}_m)$, which is the observed distribution of optimal arms. Moreover, as each agent's optimal arm played is an i.i.d. sample from $\mathcal{Q}(\mathbf{w}^\star)$, the empirical distribution $\tilde{\mathcal{Q}}$ is an unbiased estimate of $\mathcal{Q}(\mathbf{w}^\star)$.

The *inverse multi-armed bandit problem* is to recover $\mathbf{w}^\star$ given the distribution $\tilde{\mathcal{Q}}$, which allows us to identify the optimal arm. In order to achieve this, we aim to find $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, or matches it as closely as possible. Ideally, we'd want to find $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \mathcal{Q}(\mathbf{w}^\star)$,[3] but since we don't have access to $\mathcal{Q}(\mathbf{w}^\star)$, we use the unbiased estimate $\tilde{\mathcal{Q}}$ in its place.[4]

[3]Note that there might be multiple $\mathbf{w}$ such that $\mathcal{Q}(\mathbf{w}) = \mathcal{Q}(\mathbf{w}^\star)$. However, since we care only about the corresponding optimal arm, and identifiability tells us that all weight vectors with the same $\mathcal{Q}$ value have the same optimal arm, we just need to find one such weight vector.

[4]In most cases, we would have collected sufficient data such that the optimal arm corresponding to $\tilde{\mathcal{Q}}$

Since the constraint $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ is far from being convex in $\mathbf{w}$, we reformulate the problem such that the new problem is convex, and all its optimal solutions satisfy the required constraint (and vice versa). The new objective we use is the cross entropy loss between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w})$. That is, the optimization problem to solve is

$$\min_{\mathbf{w}} -\sum_{k \in A} \tilde{\mathcal{Q}}_k \log \mathcal{Q}(\mathbf{w})_k. \tag{9.1}$$

It is obvious that this objective is optimized at points with $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, if the original problem was feasible. Otherwise, it finds $\mathbf{w}$ whose $\mathcal{Q}$ is as close to $\tilde{\mathcal{Q}}$ as possible in terms of cross-entropy. Furthermore, this optimization problem is convex under a simple condition, which requires the definition of X_k as an $(m-1) \times d$ matrix with rows of the form $(\mathbf{x}_k - \mathbf{x}_j)^\intercal$, for each $j \in A \setminus \{k\}$.

Theorem 9.5.1. *Optimization problem* (9.1) *is convex if* $X_k X_k^\intercal$ *is invertible for each* $k \in A$.

The proof of the theorem appears in Appendix G.8. An exact characterization of when $X_k X_k^\intercal$ is full rank is $\operatorname{rank}(X_k X_k^\intercal) = \operatorname{rank}(X_k) = m - 1$, i.e. when X_k is full row rank. For this to be true, a necessary condition is that $d \geq m - 1$ as $\operatorname{rank}(X_k) \leq \min(d, m-1)$.[5] And under this condition, the requirement for X_k to to be full row rank is that the rows $(\mathbf{x}_k - \mathbf{x}_j)^\intercal$ are linearly independent, which is very likely to be the case, unless the feature vectors were set up adversarially. One potential scenario where the condition $d \geq m - 1$ would arise is when there are many features but feature vectors $\mathbf{x}_j$ are sparse.

As the optimization problem (9.1) is convex, we can use gradient descent to find a minimizer. And for this, we need to be able to compute the gradient accurately, which we show is possible (the calculation is given in Appendix G.9). In particular, suppose $X_k X_k^\intercal$ is invertible for each $k \in A$, and let $f(\mathbf{w})$ denote the objective function of optimization problem (9.1). Then, its gradient is given as

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = -\sum_{k \in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k \mathbf{w})} \left[\sum_{i=1}^{m-1} p_{k,i}((X_k \mathbf{w})_i) \cdot F_{k, Z_{-i} | Z_i = (X_k \mathbf{w})_i}((X_k \mathbf{w})_{-i}) \cdot X_k^{(i)} \right], \tag{9.2}$$

where F_k and p_k denote the CDF and PDF of the distribution $\mathcal{N}(0, \sigma^2 X_k X_k^\intercal)$ respectively, $F_{k, Z_{-i} | Z_i = z_i}$ is the conditional CDF of the distribution F_k given the i^{th} coordinate is z_i, $p_{k,i}$ is the PDF of the marginal distribution of this i^{th} coordinate, and $X_k^{(i)}$ denotes the i^{th} row of X_k. Note that, the conditional distribution $F_{k, Z_{-i} | Z_i = z_i}$ is also a Gaussian distribution with known parameters, and hence it can be estimated efficiently. Hence, we can use gradient descent updates defined by

$$\mathbf{w}^+ = \mathbf{w} + \alpha \sum_{k \in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k \mathbf{w})} \left[\sum_{i=1}^{m-1} p_{k,i}((X_k \mathbf{w})_i) \cdot F_{k, Z_{-i} | Z_i = (X_k \mathbf{w})_i}((X_k \mathbf{w})_{-i}) \cdot X_k^{(i)} \right],$$

coincides with the optimal arm corresponding to $\mathcal{Q}(\mathbf{w}^\star)$. It is possible that they may not coincide, but this probability goes to zero as the size of the dataset $\mathcal{D}$ increases.

[5] Intuitively, this is because X_k is taking a d-dimensional Gaussian $\boldsymbol{\eta}$ and transforming it into $(m-1)$ Gaussians via linear transformations.

where α is a suitable step size, to find an optimal solution of (9.1).

Importantly, we can also use our procedure to determine whether the optimal arm is identifiable. Given $\tilde{\mathcal{Q}}$, we solve the optimization problem (9.1) to first find a $\mathbf{w}_o$ such that $\mathcal{Q}(\mathbf{w}_o) = \tilde{\mathcal{Q}}$. Let $\mathbf{w}_o$ have the optimal arm $a_o \in A$. Now, our goal is to check if there exists any other weight $\mathbf{w}$ that has $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ but whose corresponding optimal arm is not a_o. To do this, we can build a set of convex programs, each with the exact same criterion (taking care of the $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$ requirement), but with the constraint that arm $a_i \neq a_o$ is the optimal arm (or at least beats a_o) with respect to $\mathbf{w}$. In particular, the constraint for program i could be $\mathbf{w}^\intercal \mathbf{x}_i > \mathbf{w}^\intercal \mathbf{x}_{a_o}$. As this is a simple affine constraint, solving the convex program would be very similar to running gradient descent as before. If any of these convex programs outputs an optimal solution that satisfies $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, then the problem is not identifiable, as it implies that there exist weight vectors with different optimal arms leading to the same $\tilde{\mathcal{Q}}$. On the other hand, if none of them satisfies $\mathcal{Q}(\mathbf{w}) = \tilde{\mathcal{Q}}$, we can conclude that a_o is the desired unique optimal arm.

9.5.1 Experiments

In this section, we present the empirical performance of using optimization problem (9.1) to find the ground truth weight vector $\mathbf{w}^\star$. We demonstrate this on bandit problems inspired from the counter-example from Lemma 9.4.2. The reason for this is as follows. In purely randomly generated bandit problems, the optimal arm $a^\star$ ends up being the mode of $\mathcal{Q}(\mathbf{w}^\star)$ with high probability, making the mode of $\tilde{\mathcal{Q}}$ a very good estimator of $a^\star$. This is because, for each arm a, the region $\mathcal{R}_a = \{\mathbf{w} : \mathbf{w}^\intercal \mathbf{x}_a \geq \mathbf{w}^\intercal \mathbf{x}_j \text{ for each } j\}$, corresponding to where arm a is optimal, forms a polytope, and the optimal arm's region $\mathcal{R}_{a^\star}$ contains $\mathbf{w}^\star$. Hence, as long as $\mathcal{R}_{a^\star}$ has enough volume around $\mathbf{w}^\star$, it would capture a majority of the density of the noise $\boldsymbol{\eta}$, and $a^\star$ would be the mode of the distribution $\mathcal{Q}(\mathbf{w}^\star)$. In order to avoid such "simple" instances of the problem, we consider more difficult ones inspired from our counter-example from Lemma 9.4.2.

In particular, the bandit instances we consider are as follows. There are two features ($d = 2$) and three arms $A = \{1, 2, 3\}$, and their features are defined as

$$\mathbf{x}_1 = [1, 1], \mathbf{x}_2 = [2, -\delta] \text{ and } \mathbf{x}_3 = [-\delta, 2],$$

where $\delta > 0$ is a positive constant. The ground truth weight vector is given as $\mathbf{w}^\star = [1, 1]$. Hence, for any $\delta > 0$, the optimal arm is arm 1. The noise is $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$. Such an instance is very similar to the one of Lemma 9.4.2, except that the features are not replicated to extend from two to four features, and hence the problem remains identifiable. In order to see this, recall that the regions where each arm is optimal is given by Figure 9.1. The blue, orange and green regions denote where arms 1, 2 and 3 are optimal respectively. Hence, when $\mathbf{w}^\star = [1, 1]$, the orange and green regions have equal probability, and the probability of the blue region depends on the value of σ and δ. Therefore, given the value of $\mathcal{Q}(\mathbf{w}^\star)$, one can decipher $\mathbf{w}^\star$ as follows. First, as arms 2 and 3 have the same probability, it implies that $\mathbf{w}^\star$ lies on the $w_1 = w_2$ line (because if it did not, either the orange or green would have a higher probability, depending on which side $\mathbf{w}^\star$ falls in). Next, for a given value of σ

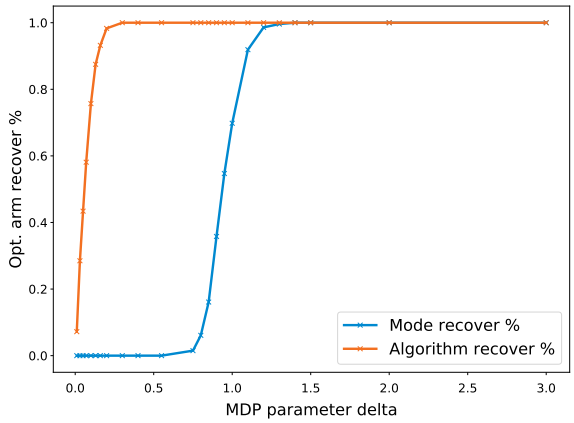


Figure 9.2: Performance as δ is varied.

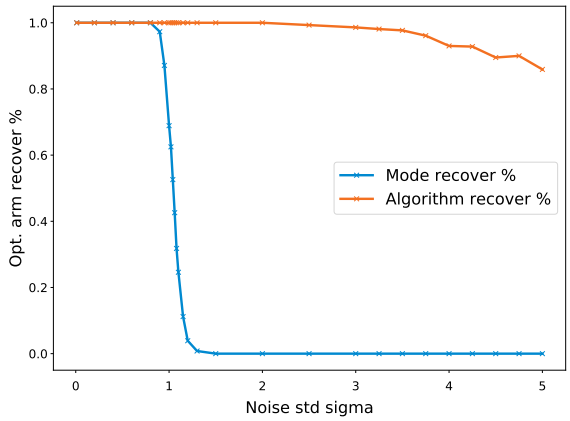


Figure 9.3: Performance as σ is varied.

and δ, any point on the $w_1 = w_2$ would lead to a unique probability of the blue region (as the noise is $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$). In particular, one could invert the value of $\mathcal{Q}(\mathbf{w}^\star)_1$ to obtain where exactly $\mathbf{w}^\star$ lies on the $w_1 = w_2$ line. Hence, the problem is identifiable (with arm 1 being the corresponding identifiable optimal arm).

Observe that when the value of δ is small enough, the blue region becomes a sliver (Figure 9.1), capturing a very small density of the noise $\boldsymbol{\eta}$, and causing arm 1 to not be the mode of $\mathcal{Q}(\mathbf{w}^\star)$. Alternatively, for a given value of δ, if σ is large enough, most of the noise's density escapes the blue region, again causing arm 1 to not be the mode of $\mathcal{Q}(\mathbf{w}^\star)$. In the following experiments, we vary both δ and σ, and show that even when the optimal arm appears negligibly in $\mathcal{Q}(\mathbf{w}^\star)$, solving optimization problem (9.1) is able to recover it.

Varying parameter δ. In the first set of experiments, we fix the noise standard deviation σ to 1, generate $n = 500$ agents according to the noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$, and vary parameter δ from 0.01 to 3. Figure 9.2 shows the percentage of times optimization problem (9.1) and the mode recover the optimal arm 1. This graph is averaged over 1000 runs. When δ is extremely close to 0, the optimal arm's region becomes a sliver and almost vanishes. Hence, small differences between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w}^\star)$ could have a substantial effect, and unless $\mathbf{w}^\star$ is numerically recovered within this sliver, the optimal arm would not be recovered. But as we move to even slightly larger values of δ, the performance of the algorithm improves substantially and it ends up recovering the optimal arm 100% of the time.

On the other hand, as δ is varied from 0 to ∞, the density of the noise $\boldsymbol{\eta}$ captured by the blue region increases continuously from 0 to that of the first quadrant. In particular, there is a point where $\mathcal{Q}(\mathbf{w}^\star)$ has probability tied across the three arms, beyond which arm 1 is always the mode (i.e. mode has 100% performance), and before which arms 2 and 3 are the modes (i.e the mode has 0% performance). This tipping point is evident from the graph and occurs around $\delta = 1$. The transition in this graph is smoother than a step function because we use the empirical mode from $\tilde{\mathcal{Q}}$ whose performance varies smoothly as the distance between probabilities of arms 1 and $\{2, 3\}$ changes.[6] Observe that the

[6]the transition would be a sharp step function if we used the mode directly from $\mathcal{Q}(\mathbf{w}^\star)$.

performance of the algorithm rises to 100% much before this tipping point, serving as evidence that it can perform really well even if the optimal arm bearly appears in the dataset. Appendix G.10.1 shows how the graph changes as δ is varied, but while the parameters are set to $\sigma \in \{0.5, 2.0\}$ or $n \in \{250, 1000\}$ instead.

Varying noise parameter σ. Here, we fix the parameter δ to 1 and generate $n = 500$ agents according to noise $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2)$, while varying the noise parameter σ from 0.01 to 5. Figure 9.3 shows the percentage of times optimization problem (9.1) and the mode recover the optimal arm 1. This graph is also averaged over 1000 runs. The results are similar in spirit to Figure 9.2. When σ is extremely large (relative to the ground truth vector $\mathbf{w}^\star = [1, 1]$), the weight space becomes less and less distinguishable w.r.t. their corresponding $\mathcal{Q}$ values. In particular, small differences between $\tilde{\mathcal{Q}}$ and $\mathcal{Q}(\mathbf{w}^\star)$ could again have a substantial effect on the corresponding optimal arms, causing a suboptimal arm to be recovered. At more reasonable levels of noise though, we can see that the algorithm recovers the optimal arm 1 100% of the time.

The mode's performance also has a similar flavor to Figure 9.2. For a given value of δ, the regions of Figure 9.1 are completely decided. When σ is close to zero, the noise is almost negligible, and hence the blue region captures most of the density of the noise $\boldsymbol{\eta}$, and the optimal arm is the mode. But as σ is varied from 0 to ∞, the density captured by this region decreases continuously from 1 to a ratio of the volumes of the regions. In particular, we again come across a point where $\mathcal{Q}(\mathbf{w}^\star)$ has probability tied across the three arms, but this time, before which arm 1 is always the mode (i.e. mode has 100% performance), and beyond which arms 2 and 3 are the modes (i.e. the mode has 0% performance). Note that, for $\sigma = 1$, this point was achieved around $\delta = 1$ (Figure 9.2). Hence, when we vary σ while fixing $\delta = 1$, the tipping point is expected to be achieved around $\sigma = 1$, which is indeed the case, as evident from Figure 9.3. Again, observe that the performance of the algorithm is still around 100% even much after this tipping point. Appendix G.10.2 shows how the graph changes as σ is varied, but while the parameters are set to $\delta \in \{0.5, 2.0\}$ or $n \in \{250, 1000\}$ instead.

9.6 Discussion

We have shown that it is possible to match the performance of the uniform mixture π^u, or that of the average agent. In Section 9.5 we then established that it is possible to learn policies from demonstrations with *superior* performance compared to the teacher, albeit under simplifying assumptions. An obvious challenge is to relax the assumptions, but this is very difficult, and we do not know of existing work that can be applied directly to our general setting. Indeed, the most relevant theoretical work is that of Syed and Schapire [SS08]. Their approach can only be applied if the sign of the reward weight is known for every feature. This is particularly problematic in our setting as some agents may consider a feature to be positive, while others consider it to be negative. A priori, it is unclear how the sign can be determined, which crucially invalidates the algorithm's theoretical guarantees. Furthermore, it is unclear under which cases the algorithm would

produce a policy with superior performance, or even if such cases exist.

We also remark that, although in the general setting we seek to compete with π^u, we are actually doing something quite different. Indeed, *ex post* (after the randomness has been instantiated) the uniform mixture π^u simply coincides with one of the individual policies. By contrast, IRL algorithms pool the feature expectations of the trajectories $\tau_1, \ldots, \tau_n$ together, and try to recover a policy that approximately matches them. Therefore, we believe that IRL algorithms do a much better job of aggregating the individual policies than π^u does, while giving almost the same optimality guarantees.

Apropos aggregation, one could make it more explicit. Specifically, suppose that we have learned (via IRL) a reward function and an optimal policy for each agent. Note that this would require a significant amount of data for each agent. Still, how should these policies be aggregated into a single policy? We can cast this as a problem of allocating public goods. A naïve approach would compute each agent's reward for each possible policy, and choose the policy that, say, maximizes the Nash social welfare [FMS18]; but this is a pipe dream, due to seemingly insurmountable computational barriers. The discovery of tractable methods for this policy aggregation problem may provide attractive alternatives to the approach presented in this paper.

Part IV

Appendix

Appendix A

Omitted Proofs for Chapter 2

A.1 Natarajan Dimension Primer

We briefly present the Natarajan dimension. For more details, we refer the reader to [SB14].

We say that a family $\mathcal{G}$ *multi-class shatters* a set of points $x_1, \ldots, x_n$ if there exist labels $y_1, \ldots y_n$ and $y'_1, \ldots, y'_n$ such that for every $i \in [n]$ we have $y_i \neq y'_i$, and for any subset $C \subset [n]$ there exists $g \in \mathcal{G}$ such that $g(x_i) = y_i$ if $i \in C$ and $g(x_i) = y'_i$ otherwise. The Natarajan dimension of a family $\mathcal{G}$ is the cardinality of the largest set of points that can be multi-class shattered by $\mathcal{G}$.

For example, suppose we have a feature map $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^q$ that maps each individual-outcome pair to a q-dimensional feature vector, and consider the family of functions that can be written as $g(x) = \arg\max_{y \in \mathcal{Y}} w^\top \Psi(x, y)$ for weight vectors $w \in \mathbb{R}^q$. This family has Natarajan dimension at most q.

For a set $S \subset \mathcal{X}$ of points, we let $\mathcal{G}\big|_S$ denote the restriction of $\mathcal{G}$ to S, which is any subset of $\mathcal{G}$ of minimal size such that for every $g \in \mathcal{G}$ there exists $g' \in \mathcal{G}\big|_S$ such that $g(x) = g'(x)$ for all $x \in S$. The size of $\mathcal{G}\big|_S$ is the number of different labelings of the sample S achievable by functions in $\mathcal{G}$. The following Lemma is the analogue of Sauer's lemma for binary classification.

Lemma A.1.1 (Natarajan)**.** *For a family $\mathcal{G}$ of Natarajan dimension d and any subset $S \subset \mathcal{X}$, we have $\left|\mathcal{G}\big|_S\right| \leq |S|^d |\mathcal{Y}|^{2d}$.*

Classes of low Natarajan dimension also enjoy the following uniform convergence guarantee.

Lemma A.1.2. *Let $\mathcal{G}$ have Natarajan dimension d and fix a loss function $\ell : \mathcal{G} \times \mathcal{X} \to [0, 1]$. For any distribution P over $\mathcal{X}$, if S is an i.i.d. sample drawn from P of size $O(\frac{1}{\epsilon^2}(d \log |\mathcal{Y}| + \log \frac{1}{\delta}))$, then with probability at least $1-\delta$ we have $\sup_{g \in \mathcal{G}} \left|\mathbb{E}_{x \sim P}[\ell(g, x)] - \frac{1}{n} \sum_{x \in S} \ell(g, x)\right| \leq \epsilon$.*

A.2 Appendix for Section 2.3

Theorem 2.3.1. *Let d be a metric on $\mathcal{X}$, P be a distribution on $\mathcal{X}$, and u be an L-Lipschitz utility function. Let S be a set of individuals such that there exists $\hat{\mathcal{X}} \subset \mathcal{X}$ with $P(\hat{\mathcal{X}}) \geq 1-\alpha$ and $\sup_{x \in \hat{\mathcal{X}}} d(x, \mathrm{NN}_S(x)) \leq \beta/(2L)$. Then for any classifier $h : S \to \Delta(\mathcal{Y})$ that is EF on S, the extension $\overline{h} : \mathcal{X} \to \Delta(\mathcal{Y})$ given by $\overline{h}(x) = h(\mathrm{NN}_S(x))$ is (α, β)-EF on P.*

Proof. Let $h : S \to \Delta(\mathcal{Y})$ be any EF classifier on S and $\overline{h} : \mathcal{X} \to \Delta(\mathcal{Y})$ be the nearest neighbor extension. Sample x and x' from P. Then, x belongs to the subset $\hat{\mathcal{X}}$ with probability at least $1-\alpha$. When this occurs, x has a neighbor within distance $\beta/(2L)$ in the sample. Using the Lipschitz continuity of u, we have $|u(x, \overline{h}(x)) - u(\mathrm{NN}_S(x), h(\mathrm{NN}_S(x)))| \leq \beta/2$. Similarly, $|u(x, \overline{h}(x')) - u(\mathrm{NN}_S(x), h(\mathrm{NN}_S(x')))| \leq \beta/2$. Finally, since $\mathrm{NN}_S(x)$ does not envy $\mathrm{NN}_S(x')$ under h, it follows that x does not envy x' by more than β under $\overline{h}$. □

Lemma A.2.1. *Suppose $\mathcal{X} \subset \mathbb{R}^q$, $d(x, x') = \|x - x'\|_2$, and let $D = \sup_{x,x' \in \mathcal{X}} d(x, x')$ be the diameter of $\mathcal{X}$. For any distribution P over $\mathcal{X}$, $\beta > 0$, $\alpha > 0$, and $\delta > 0$ there exists $\hat{\mathcal{X}} \subset \mathcal{X}$ such that $P(\hat{\mathcal{X}}) \geq 1-\alpha$ and, if S is an i.i.d. sample drawn from P of size $|S| = O(\frac{1}{\alpha}(\frac{LD\sqrt{q}}{\beta})^q(d \log \frac{LD\sqrt{q}}{\beta} + \log \frac{1}{\delta}))$, then with probability at least $1-\delta$, $\sup_{x \in \hat{\mathcal{X}}} d(x, \mathrm{NN}_S(x)) \leq \beta/(2L)$.*

Proof. Let C be the smallest cube containing $\mathcal{X}$. Since the diameter of $\mathcal{X}$ is D, the side-length of C is at most D. Let $s = \beta/(2L\sqrt{q})$ be the side-length such that a cube with side-length s has diameter $\beta/(2L)$. It takes at most $m = \lceil D/s \rceil^q$ cubes of side-length s to cover C. Let $C_1, \ldots, C_m$ be such a covering, where each C_i has side-length s.

Let C_i be any cube in the cover for which $P(C_i) > \alpha/m$. The probability that a sample of size n drawn from P does not contain a sample in C_i is at most $(1-\alpha/m)^n \leq e^{-n\alpha/m}$. Let $I = \{i \in [m] : P(C_i) \geq \alpha/m\}$. By the union bound, the probability that there exists $i \in I$ such that C_i does not contain a sample is at most $me^{-n\alpha/m}$. Setting

$$\begin{aligned} n &= \frac{m}{\alpha} \ln \frac{m}{\delta} \\ &= O\left(\frac{1}{\alpha}\left(\frac{LD\sqrt{q}}{\beta}\right)^q \left(q \log \frac{LD\sqrt{q}}{\beta} + \log \frac{1}{\delta}\right)\right) \end{aligned}$$

results in this upper bound being δ. For the remainder of the proof, assume this high probability event occurs.

Now let $\hat{\mathcal{X}} = \bigcup_{i \in I} C_i$. For each $j \notin I$, we know that $P(C_j) < \alpha/m$. Since there at most m such cubes, their total probability mass is at most α. It follows that $P(\hat{\mathcal{X}}) \geq 1-\alpha$. Moreover, every point $x \in \hat{\mathcal{X}}$ belongs to one of the cubes C_i with $i \in I$, which also contains a sample point. Since the diameter of the cubes in our cover is $\beta/(2L)$, it follows that $\mathrm{dist}(x, \mathrm{NN}_S(x)) \leq \beta/(2L)$ for every $x \in \hat{\mathcal{X}}$, as required. □

Theorem 2.3.2. *There exists a space of individuals $\mathcal{X} \subset \mathbb{R}^q$, and a distribution P over $\mathcal{X}$ such that, for every randomized algorithm $\mathcal{A}$ that extends classifiers on a sample to $\mathcal{X}$,*

there exists an L-Lipschitz utility function u such that, when a sample of individuals S of size $n = 4^q/2$ is drawn from P without replacement, there exists an EF classifier on S for which, with probability at least $1 - 2\exp(-4^q/100) - \exp(-4^q/200)$ jointly over the randomness of $\mathcal{A}$ and S, its extension by $\mathcal{A}$ is not (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$.

Proof. Let the space of individuals be $\mathcal{X} = [0,1]^q$ and the outcomes be $\mathcal{Y} = \{0,1\}$. We partition the space $\mathcal{X}$ into cubes of side length $s = 1/4$. So, the total number of cubes is $m = (1/s)^q = 4^q$. Let these cubes be denoted by $c_1, c_2, \ldots c_m$, and let their centers be denoted by $\mu_1, \mu_2, \ldots \mu_m$. Next, let P be the uniform distribution over the centers $\mu_1, \mu_2, \ldots \mu_m$. For brevity, whenever we say "utility function" in the rest of the proof, we mean "L-Lipschitz utility function."

To prove the theorem, we use Yao's minimax principle [Yao77]. Specifically, consider the following two-player zero sum game. Player 1 chooses a deterministic algorithm $\mathcal{D}$ that extends classifiers on a sample to $\mathcal{X}$, and player 2 chooses a utility function u on $\mathcal{X}$. For any subset $S \subset \mathcal{X}$, define the classifier $h_{u,S} : S \to \mathcal{Y}$ by assigning each individual in S to his favorite outcome with respect to the utility function u, i.e. $h_{u,S}(x) = \arg\max_{y \in \mathcal{Y}} u(x,y)$ for each $x \in S$, breaking ties lexicographically. Define the cost of playing algorithm $\mathcal{D}$ against utility function u as the probability over the sample S (of size $m/2$ drawn from P without replacement) that the extension of $h_{u,S}$ by $\mathcal{D}$ is not (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$. Yao's minimax principle implies that for any randomized algorithm $\mathcal{A}$, its expected cost with respect to the worst-case utility function u is at least as high as the expected cost of any distribution over utility functions that is played against the best deterministic algorithm $\mathcal{D}$ (which is tailored for that distribution). Therefore, we establish the desired lower bound by choosing a specific distribution over utility functions, and showing that the best deterministic algorithm against it has an expected cost of at least $1 - 2\exp(-m/100) - \exp(-m/200)$.

To define this distribution over utility functions, we first sample outcomes $y_1, y_2, \ldots, y_m$ i.i.d. from Bernoulli(1/2). Then, we associate each cube center μ_i with the outcome y_i, and refer to this outcome as the *favorite* of μ_i. For brevity, let $\neg y$ denote the outcome other than y, i.e. $\neg y = (1 - y)$. For any $x \in \mathcal{X}$, we define the utility function as follows. Letting c_j be the cube that x belongs to,

$$u(x, y_j) = L\left[\frac{s}{2} - \|x - \mu_j\|_\infty\right]; \quad u(x, \neg y_j) = 0. \tag{A.1}$$

See Figure A.1 for an illustration.

We claim that the utility function of Equation (A.1) is indeed L-Lipschitz with respect to any L_p norm. This is because for any cube c_i, and for any $x, x' \in c_i$, we have

$$\begin{aligned} |u(x, y_i) - u(x', y_i)| &= L\,|\|x - \mu_i\|_\infty - \|x' - \mu_i\|_\infty| \\ &\leq L\|x - x'\|_\infty \leq L\|x - x'\|_p. \end{aligned}$$

Moreover, for the other outcome, we have $u(x, \neg y_i) = u(x', \neg y_i) = 0$. It follows that u is L-Lipschitz within every cube. At the boundary of the cubes, the utility for any outcome

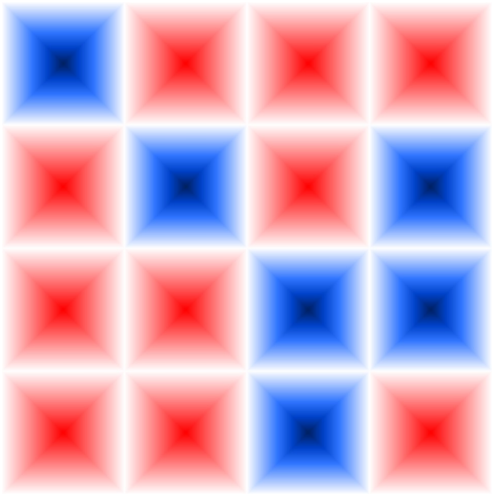


Figure A.1: Illustration of $\mathcal{X}$ and an example utility function u for $d = 2$. Red shows preference for 1, blue shows preference for 0, and darker shades correspond to more intense preference. (The gradients are rectangular to match the L_∞ norm, so, strangely enough, the misleading X pattern is an optical illusion.)

is 0, and hence u is also continuous throughout $\mathcal{X}$. Because it is piecewise Lipschitz and continuous, u must be L-Lipschitz throughout $\mathcal{X}$, with respect to any L_p norm.

Next, let $\mathcal{D}$ be an arbitrary deterministic algorithm that extends classifiers on a sample to $\mathcal{X}$. We draw the sample S of size $m/2$ from P without replacement. Consider the distribution over favorites of individuals in S. Each individual in S has a favorite that is sampled independently from Bernoulli(1/2). Hence, by Hoeffding's inequality, the fraction of individuals in S with a favorite of 0 is between $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$ with probability at least $1 - 2\exp(-m\epsilon^2)$. The same holds simultaneously for the fraction of individuals with favorite 1.

Given the sample S and the utility function u on the sample (defined by the instantiation of their favorites), consider the classifier $h_{u,S}$, which maps each individual μ_i in the sample S to his favorite y_i. This classifier is clearly EF on the sample. Consider the extension $h^{\mathcal{D}}_{u,S}$ of $h_{u,S}$ to the whole of $\mathcal{X}$ as defined by algorithm $\mathcal{D}$. Define two sets Z_0 and Z_1 by letting $Z_y = \{\mu_j \notin S \mid h^{\mathcal{D}}_{u,S}(\mu_j) = y\}$, and let y_* denote an outcome that is assigned to at least half of the out-of-sample centers, i.e., an outcome for which $|Z_{y_*}| \geq |Z_{\neg y_*}|$. Furthermore, let θ denote the fraction of out-of-sample centers assigned to y_*. Note that, since $|S| = m/2$, the number of out-of-sample centers is also exactly $m/2$. This gives us $|Z_{y_*}| = \theta\frac{m}{2}$, where $\theta \geq \frac{1}{2}$.

Consider the distribution of favorites in Z_{y_*} (these are independent from the ones in the sample since Z_{y_*} is disjoint from S). Each individual in this set has a favorite sampled independently from Bernoulli(1/2). Hence, by Hoeffding's inequality, the fraction of individuals in Z_{y_*} whose favorite is $\neg y_*$ is at least $\frac{1}{2} - \epsilon$ with probability at least $1 - \exp(-\frac{m}{2}\epsilon^2)$. We conclude that with a probability at least $1 - 2\exp(-m\epsilon^2) - \exp(-\frac{m}{2}\epsilon^2)$, the sample S and favorites (which define the utility function u) are such that: (i) the fraction of individuals in S whose favorite is $y \in \{0, 1\}$ is between $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$, and (ii) the fraction of

individuals in Z_{y_*} whose favorite is $\neg y_*$ is at least $\frac{1}{2} - \epsilon$.

We now show that for such a sample S and utility function u, $h^{\mathcal{D}}_{u,S}$ cannot be (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$. To this end, sample x and x' from P. One scenario where x envies x' occurs when (i) the favorite of x is $\neg y_*$, (ii) x is assigned to y_*, and (iii) x' is assigned to $\neg y_*$. Conditions (i) and (ii) are satisfied when x is in Z_{y_*} and his favorite is $\neg y_*$. We know that at least a $\frac{1}{2} - \epsilon$ fraction of the individuals in Z_{y_*} have the favorite $\neg y_*$. Hence, the probability that conditions (i) and (ii) are satisfied by x is at least $(\frac{1}{2} - \epsilon)|Z_{y_*}|\frac{1}{m} = (\frac{1}{2} - \epsilon)\frac{\theta}{2}$. Condition (iii) is satisfied when x' is in S and has favorite $\neg y_*$ (and hence assigned $\neg y_*$), or, if x' is in $Z_{\neg y_*}$. We know that at least a $\left(\frac{1}{2} - \epsilon\right)$ fraction of the individuals in S have the favorite $\neg y_*$. Moreover, the size of $Z_{\neg y_*}$ is $(1-\theta)\frac{m}{2}$. So, the probability that condition (iii) is satisfied by x' is at least

$$\frac{\left(\frac{1}{2} - \epsilon\right)|S| + |Z_{\neg y_*}|}{m} = \frac{1}{2}\left(\frac{1}{2} - \epsilon\right) + \frac{1}{2}(1-\theta).$$

Since x and x' are sampled independently, the probability that all three conditions are satisfied is at least

$$\left(\frac{1}{2} - \epsilon\right)\frac{\theta}{2} \cdot \left[\frac{1}{2}\left(\frac{1}{2} - \epsilon\right) + \frac{1}{2}(1-\theta)\right].$$

This expression is a quadratic function in θ, that attains its minimum at $\theta = 1$ irrespective of the value of ϵ. Hence, irrespective of $\mathcal{D}$, this probability is at least $\left[\frac{1}{2}\left(\frac{1}{2} - \epsilon\right)\right]^2$. For concreteness, let us choose ϵ to be $1/10$ (although it can be set to be much smaller). On doing so, we have that the three conditions are satisfied with probability at least $1/25$. And when these conditions are satisfied, we have $u(x, h^{\mathcal{D}}_{u,S}(x)) = 0$ and $u(x, h^{\mathcal{D}}_{u,S}(x')) = Ls/2$, i.e., x envies x' by $Ls/2 = L/8$. This shows that, when x and x' are sampled from P, with probability at least $1/25$, x envies x' by $L/8$. We conclude that with probability at least $1 - 2\exp(-m/100) - \exp(-m/200)$ jointly over the selection of the utility function u and the sample S, the extension of $h_{u,S}$ by $\mathcal{D}$ is not (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$.

To convert the joint probability into expected cost in the game, note that for two discrete, independent random variables X and Y, and for a Boolean function $\mathcal{E}(X, Y)$, it holds that

$$\Pr_{X,Y}(\mathcal{E}(X,Y) = 1) = \mathbb{E}_X\left[\Pr_Y(\mathcal{E}(X,Y) = 1)\right]. \tag{A.2}$$

Given sample S and utility function u, let $\mathcal{E}(u, S)$ be the Boolean function that equals 1 if and only if the extension of $h_{u,S}$ by $\mathcal{D}$ is not (α, β)-EF with respect to P for any $\alpha < 1/25$ and $\beta < L/8$. From Equation (A.2), $\Pr_{u,S}(\mathcal{E}(u,S) = 1)$ is equal to $\mathbb{E}_u[\Pr_S(\mathcal{E}(u,S) = 1)]$. The latter term is exactly the expected value of the cost, where the expectation is taken over the randomness of u. It follows that the expected cost of (any) $\mathcal{D}$ with respect to the chosen distribution over utilities is at least $1 - 2\exp(-m/100) - \exp(-m/200)$. □

A.3 Appendix for Section 2.4

This section is devoted to proving our main result:

Theorem 2.4.1. *Suppose $\mathcal{G}$ is a family of deterministic classifiers of Natarajan dimension d, and let $\mathcal{H} = \mathcal{H}(\mathcal{G}, m)$ for $m \in \mathbb{N}$. For any distribution P over $\mathcal{X}$, $\gamma > 0$, and $\delta > 0$, if $S = \{(x_i, x_i')\}_{i=1}^n$ is an i.i.d. sample of pairs drawn from P of size*

$$n \geq O\left(\frac{1}{\gamma^2}\left(dm^2 \log \frac{dm|\mathcal{Y}| \log(m|\mathcal{Y}|/\gamma)}{\gamma} + \log \frac{1}{\gamma}\right)\right),$$

then with probability at least $1 - \delta$, every classifier $h \in \mathcal{H}$ that is (α, β)-pairwise-EF on S is also $(\alpha + 7\gamma, \beta + 4\gamma)$-EF on P.

We start with an observation that will be required later.

Lemma A.3.1. *Let $\mathcal{G} = \{g : \mathcal{X} \to \mathcal{Y}\}$ have Natarajan dimension d. For $g_1, g_2 \in \mathcal{G}$, let $(g_1, g_2) : \mathcal{X} \to \mathcal{Y}^2$ denote the function given by $(g_1, g_2)(x) = (g_1(x), g_2(x))$ and let $\mathcal{G}^2 = \{(g_1, g_2) : g_1, g_2 \in \mathcal{G}\}$. Then the Natarajan dimension of $\mathcal{G}^2$ is at most $2d$.*

Proof. Let D be the Natarajan dimension of $\mathcal{G}^2$. Then we know that there exists a collection of points $x_1, \ldots, x_D \in \mathcal{X}$ that is shattered by $\mathcal{G}^2$, which means there are two sequences $q_1, \ldots, q_n \in \mathcal{Y}^2$ and $q_1', \ldots, q_n' \in \mathcal{Y}^2$ such that for all i we have $q_i \neq q_i'$ and for any subset $C \subset [D]$ of indices, there exists $(g_1, g_2) \in \mathcal{G}^2$ such that $(g_1, g_2)(x_i) = q_i$ if $i \in C$ and $(g_1, g_2)(x_i) = q_i'$ otherwise.

Let $n_1 = \sum_{i=1}^D \mathbb{I}\{q_{i1} \neq q_{i1}'\}$ and $n_2 = \sum_{i=1}^D \mathbb{I}\{q_{i2} \neq q_{i2}'\}$ be the number of pairs on which the first and second labels of q_i and q_i' disagree, respectively. Since none of the n pairs are equal, we know that $n_1 + n_2 \geq D$, which implies that at at least one of n_1 or n_2 must be $\geq D/2$. Assume without loss of generality that $n_1 \geq D/2$ and that $q_{i1} \neq q_{i1}'$ for $i = 1, \ldots, n_1$. Now consider any subset of indices $C \subset [n_1]$. We know there exists a pair of functions $(g_1, g_2) \in \mathcal{G}^2$ with $(g_1, g_2)(x_i)$ evaluating to q_i if $i \in C$ and q_i' if $i \notin C$. But then we have $g_1(x_i) = q_{i1}$ if $i \in C$ and $g_1(x_i) = q_{i1}'$ if $i \notin C$, and $q_{i1} \neq q_{i1}'$ for all $i \in [n_1]$. It follows that $\mathcal{G}$ shatters $x_1, \ldots, x_{n_1}$, which consists of at least $D/2$ points. Therefore, the Natarajan dimension of $\mathcal{G}^2$ is at most $2d$, as required. □

We now turn two the theorem's two main steps, presented in the following two lemmas.

Lemma A.3.2. *Let $\mathcal{H} \subset \{h : \mathcal{X} \to \Delta(\mathcal{Y})\}$ be a finite family of classifiers. For any $\gamma > 0$, $\delta > 0$, and $\beta \geq 0$ if $S = \{(x_i, x_i')\}_{i=1}^n$ is an i.i.d. sample of pairs from P of size $n \geq \frac{1}{2\gamma^2} \ln \frac{|\mathcal{H}|}{\delta}$, then with probability at least $1 - \delta$, every $h \in \mathcal{H}$ that is (α, β)-pairwise-EF on S (for any α) is also $(\alpha + \gamma, \beta)$-EF on P.*

Proof. Let $f(x, x', h) = \mathbb{I}\{u(x, h(x)) < u(x, h(x')) - \beta\}$ be the indicator that x is envious of x' by at least β under classifier h. Then $f(x_i, x_i', h)$ is a Bernoulli random variable with success probability $\mathbb{E}_{x,x' \sim P}[f(x, x', h)]$. Applying Hoeffding's inequality to any fixed hypothesis $h \in \mathcal{H}$ guarantees that $\Pr_S(\mathbb{E}_{x,x' \sim P}[f(x, x', h)] \geq \frac{1}{n}\sum_{i=1}^n f(x_i, x_i', h) + \gamma) \leq \exp(-2n\gamma^2)$. Therefore, if h is (α, β)-EF on S, then it is also $(\alpha + \gamma, \beta)$-EF on P with probability at least $1 - \exp(-2n\gamma^2)$. Applying the union bound over all $h \in \mathcal{H}$ and using the lower bound on n completes the proof. □

Next, we show that $\mathcal{H}(\mathcal{G}, m)$ can be covered by a finite subset. Since each classifier in $\mathcal{H}$ is determined by the choice of m functions from $\mathcal{G}$ and mixing weights $\eta \in \Delta_m$, we will construct finite covers of $\mathcal{G}$ and Δ_m. Our covers $\hat{\mathcal{G}}$ and $\hat{\Delta}_m$ will guarantee that for every

$g \in \mathcal{G}$, there exists $\hat{g} \in \hat{\mathcal{G}}$ such that $\Pr_{x\sim P}(g(x) \neq \hat{g}(x)) \leq \gamma/m$. Similarly, for any mixing weights $\eta \in \Delta_m$, there exists $\hat{\eta} \in \Delta_m$ such that $\|\eta - \hat{\eta}\|_1 \leq \gamma$. If $h \in \mathcal{H}(\mathcal{G}, m)$ is the mixture of $g_1, \ldots, g_m$ with weights η, we let $\hat{h}$ be the mixture of $\hat{g}_1, \ldots, \hat{g}_m$ with weights $\hat{\eta}$. This approximation has two sources of error: first, for a random individual $x \sim P$, there is probability up to γ that at least one $g_i(x)$ will disagree with $\hat{g}_i(x)$, in which case h and $\hat{h}$ may assign completely different outcome distributions. Second, even in the high-probability event that $g_i(x) = \hat{g}_i(x)$ for all $i \in [m]$, the mixing weights are not identical, resulting in a small perturbation of the outcome distribution assigned to x.

Lemma A.3.3. *Let $\mathcal{G}$ be a family of deterministic classifiers with Natarajan dimension d, and let $\mathcal{H} = \mathcal{H}(\mathcal{G}, m)$ for some $m \in \mathbb{N}$. For any $\gamma > 0$, there exists a subset $\hat{\mathcal{H}} \subset \mathcal{H}$ of size $O\big(\frac{(dm|\mathcal{Y}|^2 \log(m|\mathcal{Y}|/\gamma))^{dm}}{\gamma^{(d+1)m}}\big)$ such that for every $h \in \mathcal{H}$ there exists $\hat{h} \in \hat{\mathcal{H}}$ satisfying:*

1. $\Pr_{x\sim P}(\|h(x) - \hat{h}(x)\|_1 > \gamma) \leq \gamma$.
2. *If S is an i.i.d. sample of individuals of size $O(\frac{m^2}{\gamma^2}(d \log|\mathcal{Y}| + \log\frac{1}{\delta}))$ then w.p. $\geq 1-\delta$, we have $\|h(x) - \hat{h}(x)\|_1 \leq \gamma$ for all but a 2γ-fraction of $x \in S$.*

Proof. As described above, we begin by constructing finite covers of Δ_m and $\mathcal{G}$. First, let $\hat{\Delta}_m \subset \Delta_m$ be the set of distributions over $[m]$ where each coordinate is a multiple of γ/m. Then we have $|\hat{\Delta}_m| = O((\frac{m}{\gamma})^m)$ and for every $p \in \Delta_m$, there exists $q \in \hat{\Delta}_m$ such that $\|p - q\|_1 \leq \gamma$.

In order to find a small cover of $\mathcal{G}$, we use the fact that it has low Natarajan dimension. This implies that the number of effective functions in $\mathcal{G}$ when restricted to a sample S' grows only polynomially in the size of S'. At the same time, if two functions in $\mathcal{G}$ agree on a large sample, they will also agree with high probability on the distribution.

Formally, let S' be an i.i.d. sample drawn from P of size $O(\frac{m^2}{\gamma^2} d \log|\mathcal{Y}|)$, and let $\hat{\mathcal{G}} = \mathcal{G}\big|_{S'}$ be any minimal subset of $\mathcal{G}$ that realizes all possible labelings of S' by functions in $\mathcal{G}$. We now argue that with probability 0.99, for every $g \in \mathcal{G}$ there exists $\hat{g} \in \hat{\mathcal{G}}$ such that $\Pr_{x\sim P}(g(x) \neq \hat{g}(x)) \leq \gamma/m$. For any pair of functions $g, g' \in \mathcal{G}$, let $(g, g') : \mathcal{X} \to \mathcal{Y}^2$ be the function given by $(g, g')(x) = (g(x), g'(x))$, and let $\mathcal{G}^2 = \{(g, g') : g, g' \in \mathcal{G}\}$. The Natarajan dimension of $\mathcal{G}^2$ is at most $2d$ by Lemma A.3.1. Moreover, consider the loss $c : \mathcal{G}^2 \times \mathcal{X} \to \{0, 1\}$ given by $c(g, g', x) = \mathbb{I}\{g(x) \neq g'(x)\}$. Applying Lemma A.1.2 with the chosen size of $|S'|$ ensures that with probability at least 0.99 every pair $(g, g') \in \mathcal{G}^2$ satisfies

$$\left| \mathop{\mathbb{E}}_{x\sim P}[c(g, g', x)] - \frac{1}{|S'|} \sum_{x \in S'} c(g, g', x) \right| \leq \frac{\gamma}{m}.$$

By the definition of $\hat{\mathcal{G}}$, for every $g \in \mathcal{G}$, there exists $\hat{g} \in \hat{\mathcal{G}}$ for which $c(g, \hat{g}, x) = 0$ for all $x \in S'$, which implies that $\Pr_{x\sim P}(g(x) \neq \hat{g}(x)) \leq \gamma/m$.

Using Lemma A.1.1 to bound the size of $\hat{\mathcal{G}}$, we have that

$$|\hat{\mathcal{G}}| \leq |S'|^d |\mathcal{Y}|^{2d} = O\left(\left(\frac{m^2}{\gamma^2} d|\mathcal{Y}|^2 \log|\mathcal{Y}|\right)^d\right).$$

Since this construction succeeds with non-zero probability, we are guaranteed that such a set $\hat{\mathcal{G}}$ exists. Finally, by an identical uniform convergence argument, it follows that if S

is a fresh i.i.d. sample of the size given in Item 2 of the lemma's statement, then, with probability at least $1-\delta$, every g and $\hat{g}$ will disagree on at most a $2\gamma/m$-fraction of S, since they disagree with probability at most γ/m on P.

Next, let $\hat{\mathcal{H}} = \{h_{\vec{g},\eta} : \vec{g} \in \hat{G}^m, \eta \in \hat{\Delta}_m\}$ be the same family as $\mathcal{H}$, except restricted to choosing functions from $\hat{\mathcal{G}}$ and mixing weights from $\hat{\Delta}_m$. Using the size bounds above and the fact that $\binom{N}{m} = O((\frac{N}{m})^m)$, we have that

$$|\hat{\mathcal{H}}| = \binom{|\hat{\mathcal{G}}|}{m} \cdot |\hat{\Delta}_m| = O\left(\frac{(dm^2|\mathcal{Y}|^2 \log(m|\mathcal{Y}|/\gamma))^{dm}}{\gamma^{(2d+1)m}}\right).$$

Suppose that h is the mixture of $g_1, \ldots, g_m \in \mathcal{G}$ with weights $\eta \in \Delta_m$. Let $\hat{g}_i$ be the approximation to g_i for each i, let $\hat{\eta} \in \hat{\Delta}_m$ be such that $\|\eta - \hat{\eta}\|_1 \leq \gamma$, and let $\hat{h}$ be the random mixture of $\hat{g}_1, \ldots, \hat{g}_m$ with weights $\hat{\eta}$. For an individual x drawn from P, we have $g_i(x) \neq \hat{g}_i(x)$ with probability at most γ/m, and therefore they all agree with probability at least $1-\gamma$. When this event occurs, we have $\|h(x) - \hat{h}(x)\|_1 \leq \|\eta - \hat{\eta}\|_1 \leq \gamma$.

The second part of the claim follows by similar reasoning, using the fact that for the given sample size $|S|$, with probability at least $1-\delta$, every $g \in \mathcal{G}$ disagrees with its approximation $\hat{g} \in \hat{\mathcal{G}}$ on at most a $2\gamma/m$-fraction of S. This means that $\hat{g}_i(x) = g_i(x)$ for all $i \in [m]$ on at least a $(1-2\gamma)$-fraction of the individuals x in S. For these individuals, $\|h(x) - \hat{h}(x)\|_1 \leq \|\eta - \hat{\eta}\|_1 \leq \gamma$. □

Combining the generalization guarantee for finite families given in Lemma A.3.2 with the finite approximation given in Lemma A.3.3, we are able to show that envy-freeness also generalizes for $\mathcal{H}(\mathcal{G}, m)$.

Proof of Theorem 2.4.1. Let $\hat{\mathcal{H}}$ be the finite approximation to $\mathcal{H}$ constructed in Lemma A.3.3. If the sample is of size $|S| = O(\frac{1}{\gamma^2}(dm \log(dm|\mathcal{Y}| \log|\mathcal{Y}|/\gamma) + \log\frac{1}{\delta}))$, we can apply Lemma A.3.2 to this finite family, which implies that for any $\beta' \geq 0$, with probability at least $1-\delta/2$ every $\hat{h} \in \hat{\mathcal{H}}$ that is (α', β')-pairwise-EF on S (for any α') is also $(\alpha'+\gamma, \beta')$-EF on P. We apply this lemma with $\beta' = \beta + 2\gamma$. Moreover, from Lemma A.3.3, we know that if $|S| = O(\frac{m^2}{\gamma^2}(d \log|\mathcal{Y}| + \log\frac{1}{\delta}))$, then with probability at least $1-\delta/2$, for every $h \in \mathcal{H}$, there exists $\hat{h} \in \hat{\mathcal{H}}$ satisfying $\|h(x) - \hat{h}(x)\|_1 \leq \gamma$ for all but a 2γ-fraction of the individuals in S. This implies that on all but at most a 4γ-fraction of the pairs in S, h and $\hat{h}$ satisfy this inequality for both individuals in the pair. Assume these high probability events occur. Finally, from Item 1 of the lemma we have that $\Pr_{x_1,x_2\sim P}(\max_{i=1,2} \|h(x_i) - \hat{h}(x_i)\|_1 > \gamma) \leq 2\gamma$.

Now let $h \in \mathcal{H}$ be any classifier that is (α, β)-pairwise-EF on S. Since the utilities are in $[0,1]$ and $\max_{x=x_i,x_i'} \|h(x) - \hat{h}(x)\|_1 \leq \gamma$ for all but a 4γ-fraction of the pairs in S, we know that $\hat{h}$ is $(\alpha+4\gamma, \beta+2\gamma)$-pairwise-EF on S. Applying the envy-freeness generalization guarantee (Lemma A.3.2) for $\hat{\mathcal{H}}$, it follows that $\hat{h}$ is also $(\alpha+5\gamma, \beta+2\gamma)$-EF on P. Finally, using the fact that

$$\Pr_{x_1,x_2\sim P}\left(\max_{i=1,2} \|h(x_i) - \hat{h}(x_i)\|_1 > \gamma\right) \leq 2\gamma,$$

it follows that h is $(\alpha+7\gamma, \beta+4\gamma)$-EF on P. □

It is worth noting that the (exponentially large) approximation $\hat{\mathcal{H}}$ is only used in the generalization analysis; importantly, an ERM algorithm need not construct it.

A.4 Appendix for Section 2.5

Here we describe details of the transformation of the optimization problem from (2.2) to (2.4). Firstly, softening constraints of (2.2) with slack variables, we obtain

$$\begin{aligned}\min_{g_k\in\mathcal{G},\xi\in\mathbb{R}^{n\times n}_{\geq 0}} \quad & \sum_{i=1}^{n} L(x_i, g_k(x_i)) + \lambda \sum_{i\neq j} \xi_{ij} \\ \text{s.t.} \quad & USF_{ii}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_i)) \geq USF_{ij}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_j)) - \xi_{ij} \quad \forall (i,j).\end{aligned}$$

Here, ξ_{ij} basically captures how much i envies j under the selected assignments (note that, ξ_{ij} is 0 if the pair is non-envious, so that the algorithm does not go increasing negative envy at the cost of positive envy for someone else). Plugging in optimal values of the slack variables, we obtain

$$\begin{aligned}\min_{g_k\in\mathcal{G}} \quad & \sum_{i=1}^{n} L(x_i, g_k(x_i)) \\ & + \lambda \sum_{i\neq j} \max\left(USF_{ij}^{(k-1)} + \tilde{\eta}_k u(x_i, g_k(x_j)) - USF_{ii}^{(k-1)} - \tilde{\eta}_k u(x_i, g_k(x_i)), 0\right). \quad \text{(A.3)}\end{aligned}$$

Next, we perform convex relaxation of different components of this objective function. For this, let's observe the term $L(x_i, g_k(x_i))$. And, let $\vec{w}$ denote the parameters of g_k. By definition, we have

$$w_{g_k(x_i)}^\top x_i \geq w_{y'}^\top x_i$$

for any $y' \in \mathcal{Y}$. This implies that

$$\begin{aligned}L(x_i, g_k(x_i)) &\leq L(x_i, g_k(x_i)) + w_{g_k(x_i)}^\top x_i - w_{y'}^\top x_i \\ &\leq \max_{y\in\mathcal{Y}} \left\{L(x_i, y) + w_y^\top x_i - w_{y'}^\top x_i\right\},\end{aligned}$$

giving us a convex upper bound on the loss $L(x_i, g_k(x_i))$. As this holds for any $y' \in \mathcal{Y}$, we choose $y' = y_i$ as defined in the main body, since it leads to the lowest achievable loss value. Therefore, we have

$$L(x_i, g_k(x_i)) \leq \max_{y\in\mathcal{Y}} \left\{L(x_i, y) + w_y^\top x_i - w_{y_i}^\top x_i\right\}.$$

This right hand side is basically an upper bound which apart from encouraging $\vec{w}$ to have the highest dot product with x_i at y_i, also penalizes if the margin by which this is higher is not enough (where the margin depends on other losses $L(x_i, y)$). This surrogate loss is very similar to multi-class support vector machines. We perform similar relaxations for the

other two components of the objective function. In particular, for the $u(x_i, g_k(x_i))$ term, we have

$$-u(x_i, g_k(x_i)) \leq \max_{y \in \mathcal{Y}} \left\{ -u(x_i, y) + w_y^\top x_i - w_{b_i}^\top x_i \right\},$$

where b_i is as defined in the main body. Finally, for the remaining term, we have

$$u(x_i, g_k(x_j)) \leq \max_{y \in \mathcal{Y}} \left\{ u(x_i, y) + w_y^\top x_j - w_{s_i}^\top x_j \right\},$$

where s_i is as defined in the main body[1]. On plugging in the convex surrogates of all three terms in Equation (A.3), we obtain the optimization problem (2.4).

[1]Note that, instead of using s_i, an alternative to use in this equation is b_j. In particular, for a pair (i, j), using s_i encourages the assignment to give i their favorite outcome while j the outcome that i likes the least (and hence causing i to envy j as less as possible), while using b_j encourages the assignment to give both i and j their favorite outcomes (pushing the assignment to just give everyone their favorite outcomes).

Appendix B

Omitted Proofs and Results for Chapter 3

B.1 Voting Rules

This appendix provides additional background on social choice theory. It is not required to understand the rest of the paper, but may be helpful in putting our results in context.

B.1.1 Examples of Anonymous Voting Rules

One class of anonymous voting rules uses the positions of the individual alternatives in order to determine the winners. These rules, collectively called *positional scoring rules*, are defined by a scoring vector $\mathbf{s}$ such that $s_1 \geq s_2 \geq \cdots \geq s_m \geq 0$. Given a vote σ, the score of alternative $a \in A$ in σ is the score of its position in σ, i.e., $s_{\sigma(a)}$. Given an anonymous vote profile $\boldsymbol{\pi}$, the score of an alternative is its overall score in the rankings of $\boldsymbol{\pi}$, that is,

$$s\text{-score}_{\boldsymbol{\pi}}(a) \triangleq \sum_{\sigma \in \mathcal{L}(A)} \pi_\sigma s_{\sigma(a)}.$$

A *deterministic positional scoring rule* chooses the alternative with the highest score, i.e., $f(\boldsymbol{\pi}) = \mathbf{e}_{a^*}$, where $a^* \in \arg\max_{a \in A} s\text{-score}_{\boldsymbol{\pi}}(a)$ (tie breaking may be needed). On the other hand, a *randomized positional scoring rule* chooses each alternative with probability proportional to its score, i.e., $f(\boldsymbol{\pi})_a \propto s\text{-score}_{\boldsymbol{\pi}}(a)$ for all $a \in A$. Examples of positional scoring rules include *plurality* with $\mathbf{s} = (1, 0, \ldots, 0)$, *veto* with $\mathbf{s} = (1, \ldots, 1, 0)$, and *Borda* with $\mathbf{s} = (m-1, m-2, \ldots, 0)$.

Another class of anonymous voting rules uses pairwise comparisons between the alternatives to determine the winners. We are especially interested in the *Copeland* rule, which assigns a score to each alternative based on the number of pairwise majority contests it wins. In an anonymous vote profile $\boldsymbol{\pi}$, we denote by $a >_{\boldsymbol{\pi}} b$ the event that a beats b in a pairwise competition, i.e., a is preferred to b in rankings in $\boldsymbol{\pi}$ that collectively have more than half the weight. More formally, $\sum_{\sigma \in \mathcal{L}(A)} \pi_\sigma \mathbb{1}_{(a \succ_\sigma b)} > 1/2$. We also write $a =_{\boldsymbol{\pi}} b$ if they

are tied, i.e., $\sum_{\sigma \in \mathcal{L}(A)} \pi_\sigma \mathbb{1}_{(a \succ_\sigma b)} = 1/2$. The Copeland score[1] of an alternative is defined by

$$\text{C-score}_{\boldsymbol{\pi}}(a) \triangleq |\{b \in A \mid a >_{\boldsymbol{\pi}} b\}| + \frac{1}{2} \cdot |\{b \in A \mid a =_{\boldsymbol{\pi}} b\}| .$$

The *deterministic Copeland rule* chooses the alternative that has the highest Copeland score (possibly breaking ties), and the *randomized Copeland rule* chooses each alternative with probability proportional to its Copeland score.

These notations allow us to formally define the notion of Condorcet consistency (informally introduced in Section 3.5). We say that $a \in A$ is a *Condorcet winner* in the vote profile $\boldsymbol{\pi}$ if $a >_{\boldsymbol{\pi}} b$ for all $b \in A \setminus \{a\}$. A voting rule is *Condorcet consistent* if it selects a Condorcet winner whenever one exists in the given vote profile. Note that the Copeland score of a Condorcet winner is $m - 1$, whereas the Copeland score of any other alternative must be strictly smaller, so a Condorcet winner (if one exists) indeed has maximum Copeland score.

B.1.2 Strategyproofness, More Formally

An anonymous deterministic voting rule f is called *strategyproof* if for any voter $i \in [n]$, any two vote profiles $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ for which $\sigma_j = \sigma'_j$ for all $j \neq i$, and any weight vector $\mathbf{w}$, it holds that either $a = a'$ or $a \succ_{\sigma_i} a'$, where a and a' are the winning alternatives in $f(\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}})$ and $f(\boldsymbol{\pi}_{\boldsymbol{\sigma}',\mathbf{w}})$ respectively. In words, whenever a voter reports σ'_i instead of σ_i, the outcome does not improve according to the true ranking σ_i.

To extend this definition to randomized rules, we require some additional definitions. Given a *loss function* over the alternatives denoted by a vector $\boldsymbol{\ell} \in [0,1]^m$, the expected loss of the alternative chosen by the rule f under an anonymous vote profile $\boldsymbol{\pi}$ is

$$L_f(\boldsymbol{\pi}, \boldsymbol{\ell}) \triangleq \mathbb{E}_{a \sim f(\boldsymbol{\pi})}[\ell_a] = f(\boldsymbol{\pi}) \cdot \boldsymbol{\ell}.$$

The higher the loss, the worse the alternative. We say that the loss function $\boldsymbol{\ell}$ is *consistent* with vote $\sigma \in \mathcal{L}(A)$ if for all $a, b \in A$, $a \succ_\sigma b \Leftrightarrow \ell_a < \ell_b$. An anonymous randomized rule f is *strategyproof* if for any voter $i \in [n]$, any two vote profiles $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ for which $\sigma_j = \sigma'_j$ for all $j \neq i$, any weight vector $\mathbf{w}$, and any loss function $\boldsymbol{\ell}$ that is consistent with σ_i, we have $L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}, \boldsymbol{\ell}) \leq L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}',\mathbf{w}}, \boldsymbol{\ell})$.

As noted in Section 3.2.1, randomized positional scoring rules, and the randomized Copeland rule, are known to be strategyproof. To see why they satisfy Gibbard's necessary condition (Proposition 3.2.1), a randomized positional scoring rule with score vector $\mathbf{s}$ is a distribution with probabilities proportional to $s_1, \ldots, s_m$ over anonymous unilateral rules $g_1, \ldots, g_m$, where each g_i corresponds to the function $h_i(\sigma)$ that returns the alternative ranked at position i of σ. Similarly, the randomized Copeland rule is a uniform distribution over duples $g_{a,b}$ for any two different $a, b \in A$, where $g_{a,b}(\boldsymbol{\pi}) = \mathbf{e}_a$ if $a >_{\boldsymbol{\pi}} b$, $g_{a,b}(\boldsymbol{\pi}) = \mathbf{e}_b$ if $b >_{\boldsymbol{\pi}} a$, and $(g_{a,b}(\boldsymbol{\pi}))_a = (g_{a,b}(\boldsymbol{\pi}))_b = 1/2$ if $a =_{\boldsymbol{\pi}} b$.

[1] Some refer to this variant of Copeland as Copeland$_{1/2}$ [FHS08].

B.2 Proof of Theorem 3.5.3

Let f be a distribution over unilaterals $g_1, \dots, g_k$ with corresponding probabilities $q_1, \dots, q_k$. Also, let $h_j : \mathcal{L}(A) \to A$ denote the function corresponding to g_j, for $j \in [k]$. We first prove Equation (3.5). For ease of exposition we suppress t in the notations, when it is clear from the context. Furthermore, let $\boldsymbol{\pi}^i = \boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}$. It holds that

$$\begin{aligned}
\mathbb{E}_{i \sim \mathbf{p}^t}\left[f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i})\right] &= \sum_{i=1}^{n} p_i^t f(\boldsymbol{\pi}^i) \\
&= \sum_{i=1}^{n} p_i^t \sum_{j=1}^{k} q_j \sum_{\tau \in \mathcal{L}(A)} \pi_\tau^i \mathbf{e}_{h_j(\tau)} \\
&= \sum_{i=1}^{n} p_i^t \sum_{j=1}^{k} q_j \mathbf{e}_{h_j(\sigma_i)},
\end{aligned}$$

where the last equality follows by the fact that $\pi_{\sigma_i}^i = 1$ and $\pi_\tau^i = 0$ for any $\tau \neq \sigma_i$. Moreover, let $\boldsymbol{\pi} = \boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{p}^t}$, then

$$\begin{aligned}
f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{p}^t}) &= \sum_{j=1}^{k} q_j \sum_{\tau \in \mathcal{L}(A)} \pi_\tau \mathbf{e}_{h_j(\tau)} \\
&= \sum_{j=1}^{k} q_j \sum_{\tau \in \mathcal{L}(A)} \mathbf{e}_{h_j(\tau)} \sum_{i=1}^{n} p_i^t \mathbb{1}_{(\sigma_i = \tau)} \\
&= \sum_{i=1}^{n} p_i^t \sum_{j=1}^{k} q_j \mathbf{e}_{h_j(\sigma_i)}.
\end{aligned}$$

Now that we have established Equation (3.5), we use it to conclude that

$$\begin{aligned}
&\sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{p}^t}, \boldsymbol{\ell}^t) - \min_{i \in [n]} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t) \\
&\quad = \mathbb{E}\left[\sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t) - \min_{i \in [n]} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t)\right],
\end{aligned}$$

where the expectation is taken over choice of $i \sim \mathbf{p}^t$ for all t. Therefore, the deterministic weighting schemes that use weight vector $\mathbf{p}^t$ achieve the same regret bounds as those established in Theorems 3.4.1 and 3.4.2. □

B.3 Proof of Theorem 3.5.5

We start by proving the following technical lemma.

Lemma B.3.1. *Let $x_1, x_2, \cdots x_n$ be n real numbers such that $x_i \geq x_{i+1}$ for all $i \in [n-1]$, and denote $S = \sum_{i=1}^{n} x_i$. Then for any $j \in [n]$, $\sum_{i=1}^{j} x_i \geq j\frac{S}{n}$.*

Proof. Assume for the sake of contradiction that there exists $j \in [n-1]$ such that $\sum_{i=1}^{j} x_i < j\frac{S}{n}$. It follows that there is $i \in [j]$ such that $x_i < \frac{S}{n}$. In addition, it must be the case that $\sum_{i=j+1}^{n} x_i > (n-j)\frac{S}{n}$, which implies that there is $i' \in \{j+1, \ldots, n\}$ such that $x_{i'} > \frac{S}{n}$. This contradicts the fact that $x_i \geq x_{i'}$. □

Proof of Theorem 3.5.5. Fix an arbitrary deterministic weighting scheme. We will show that the loss of this weighting scheme is strictly higher than the average loss of the voters (for appropriately chosen vote profiles and loss functions) at every time step t, which directly leads to linear regret.

Consider an arbitrary time step $t \leq T$, and let $\mathbf{w}^t$ denote the weights chosen by the weighting scheme. To construct the vote profile $\boldsymbol{\sigma}^t$, the adversary first partitions the voters into two sets N_1^t and N_2^t, as follows: It sorts the weights $\mathbf{w}^t$ in non-increasing order, and then it adds voters to N_1^t by their sorted weight (largest to smallest) until

$$W_1^t \triangleq \sum_{i \in N_1^t} w_i^t > \frac{1}{2}\|\mathbf{w}^t\|_1,$$

that is, until the voters in N_1^t have more than half the total weight. The remaining voters form set N_2^t.

Now, let $\tau^{x,y} \in \mathcal{L}(A)$ denote a ranking that places x at the top (i.e., $\tau^{x,y}(x) = 1$) and y in second place (i.e., $\tau^{x,y}(y) = 2$). Let a and b be two alternatives such that $f(\mathbf{e}_{\tau^{b,a}})_b - f(\mathbf{e}_{\tau^{b,a}})_a \geq f(\mathbf{e}_{\tau^{a,b}})_a - f(\mathbf{e}_{\tau^{a,b}})_b$, i.e., the gap between the probabilities of picking the top two alternatives in $\mathbf{e}_{\tau^{b,a}}$ is at least the corresponding gap in $\mathbf{e}_{\tau^{a,b}}$. The adversary sets the vote profile $\boldsymbol{\sigma}^t$ such that $\sigma_i^t = \tau^{a,b}$ for all $i \in N_1^t$ and $\sigma_i^t = \tau^{b,a}$ for all $i \in N_2^t$. Also, it sets the loss function $\boldsymbol{\ell}^t$ to be $\ell_a^t = 1$, $\ell_b^t = 0$, and $\ell_x^t = 1/2$ for all $x \in A \setminus \{a, b\}$.

Observe that for all $i \in N_1^t$, $a \succ_{\sigma_i} x$ for all $x \in A \setminus \{a\}$. Since the total weight of voters in N_1^t is more than $1/2$, a is a Condorcet winner in $\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}$. Therefore, because f is probabilistically Condorcet consistent with gap $\delta(m)$, it holds that

$$f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_a \geq f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_b + \delta(m).$$

It follows that the loss of the weighting scheme is

$$\begin{aligned} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}, \boldsymbol{\ell}^t) &= 1 \cdot f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_a \\ &\quad + \frac{1}{2} \cdot (1 - f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_a - f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_b) \\ &= \frac{1}{2} + \frac{1}{2}\left(f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_a - f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t})_b\right) \\ &\geq \frac{1}{2} + \frac{1}{2}\delta(m). \end{aligned} \tag{B.1}$$

Similarly, the loss of voter i is

$$\begin{aligned} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t) &= L_f(\mathbf{e}_{\sigma_i^t}, \boldsymbol{\ell}^t) \\ &= \frac{1}{2} + \frac{1}{2}\left(f(\mathbf{e}_{\sigma_i^t})_a - f(\mathbf{e}_{\sigma_i^t})_b\right). \end{aligned} \tag{B.2}$$

Let $\mathbf{q}^1$ denote $f(\mathbf{e}_{\tau^{a,b}})$, i.e. the distribution over the alternatives for the votes of voters in N_1^t, and let $\mathbf{q}^2$ denote $f(\mathbf{e}_{\tau^{b,a}})$, i.e. the distribution over the alternatives for the votes of voters in N_2^t. Using these notations and Equation (B.2), the loss of a voter $i \in N_1^t$ is

$$L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}, \boldsymbol{\ell}^t) = \frac{1}{2} + \frac{1}{2}\left(q_a^1 - q_b^1\right),$$

and the loss of a voter $i \in N_2^t$ is

$$L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}, \boldsymbol{\ell}^t) = \frac{1}{2} + \frac{1}{2}\left(q_a^2 - q_b^2\right) = \frac{1}{2} - \frac{1}{2}(q_b^2 - q_a^2).$$

Hence, the average loss over all voters is

$$\begin{aligned} L_{avg}^t &= \frac{|N_1^t|\left(\frac{1}{2} + \frac{1}{2}\left(q_a^1 - q_b^1\right)\right) + (n - |N_1^t|)\left(\frac{1}{2} - \frac{1}{2}(q_b^2 - q_a^2)\right)}{n} \\ &= \frac{1}{2} + \frac{1}{2n}\left(|N_1^t|(q_a^1 - q_b^1) - (n - |N_1^t|)(q_b^2 - q_a^2)\right). \end{aligned}$$

But we chose a and b such that $q_a^1 - q_b^1 \leq q_b^2 - q_a^2$. We conclude that

$$\begin{aligned} L_{avg}^t &\leq \frac{1}{2} + \frac{1}{2n}\left(|N_1^t|(q_b^2 - q_a^2) - (n - |N_1^t|)(q_b^2 - q_a^2)\right) \\ &= \frac{1}{2} + \frac{1}{2}(q_b^2 - q_a^2)\frac{(2|N_1^t| - n)}{n}. \end{aligned} \tag{B.3}$$

Our goal is to derive an upper bound on the expression $\frac{1}{2}(q_b^2 - q_a^2)\frac{(2|N_1^t| - n)}{n}$. Specifically, we wish to prove that

$$\frac{1}{2}(q_b^2 - q_a^2)\frac{(2|N_1^t| - n)}{n} \leq \frac{\delta(m)}{3}. \tag{B.4}$$

We do this by examining two cases.

Case 1: $W_1^t \geq \left(\frac{1}{2} + \frac{\delta(m)}{3}\right)\|\mathbf{w}^t\|_1$**.** Informally, this is the case when the weights of N_1^t overshot $\|\mathbf{w}^t\|_1/2$ by a fraction of at least $\delta(m)/3$. This means that the last voter added to N_1^t has a weight of at least $W_1^t - \frac{\|\mathbf{w}^t\|_1}{2}$. Since the weights were added in non-increasing order, it follows that each voter in N_1^t has a weight of at least $W_1^t - \frac{\|\mathbf{w}^t\|_1}{2}$. Therefore,

$$\begin{aligned} W_1^t &= \sum_{i \in N_1^t} w_i^t \geq \sum_{i \in N_1^t}\left(W_1^t - \frac{\|\mathbf{w}^t\|_1}{2}\right) \\ &= |N_1^t|\left(W_1^t - \frac{\|\mathbf{w}^t\|_1}{2}\right), \end{aligned}$$

or equivalently,

$$|N_1^t| \leq \frac{1}{1 - \frac{\|\mathbf{w}^t\|_1}{2W_1^t}}. \tag{B.5}$$

We have also assumed that $\frac{W_1^t}{\|\mathbf{w}^t\|_1} \geq \left(\frac{1}{2} + \frac{\delta(m)}{3}\right)$. Using Equation (B.5), we obtain

$$|N_1^t| \leq \frac{1}{1 - \frac{1}{1+\frac{2\delta(m)}{3}}} = \frac{3}{2\delta(m)} + 1. \tag{B.6}$$

Let us now examine the expression on the left-hand side of Equation (B.4). Note that b is a Condorcet winner in $\mathbf{e}_{\tau^{b,a}}$. Hence, $q_b^2 \geq q_a^2 + \delta(m)$, and, in particular, $q_b^2 - q_a^2 > 0$. In addition, we have assumed that $n \geq 2(\frac{3}{2\delta(m)} + 1)$, which implies (by Equation (B.6)) that $n \geq 2|N_1^t|$. It follows that

$$\frac{1}{2}(q_b^2 - q_a^2)\frac{(2|N_1^t| - n)}{n} \leq 0 \leq \frac{\delta(m)}{3},$$

thereby establishing Equation (B.4) for this case.

Case 2: $W_1^t < \left(\frac{1}{2} + \frac{\delta(m)}{3}\right)\|\mathbf{w}^t\|_1$**.** Since N_1^t contains voters who have the largest $|N_1^t|$ weights, Lemma B.3.1 implies that

$$W_1^t = \sum_{i \in N_1^t} w_i^t \geq |N_1^t| \frac{\|\mathbf{w}^t\|_1}{n}.$$

We have also assumed that $W_1^t < (\frac{1}{2} + \frac{\delta(m)}{3})\|\mathbf{w}^t\|_1$. Combining the last two inequalities, we obtain

$$|N_1^t| < n\left(\frac{1}{2} + \frac{\delta(m)}{3}\right). \tag{B.7}$$

Let us examine, once again, the left-hand side of Equation (B.4). Recall that $q_b^2 - q_a^2 > 0$, because b is a Condorcet winner in $\tau^{b,a}$. So, if $2|N_1^t| - n \leq 0$, then Equation (B.4) clearly holds, as in Case 1. And if $2|N_1^t| - n > 0$, the equation also holds, because

$$\begin{aligned}\frac{1}{2}(q_b^2 - q_a^2)\frac{(2|N_1^t| - n)}{n} &\leq \frac{1}{2} \cdot 1 \cdot \frac{(2|N_1^t| - n)}{n} \\ &= \frac{|N_1^t|}{n} - \frac{1}{2} \\ &< \frac{\delta(m)}{3},\end{aligned}$$

where the last inequality follows from Equation (B.7).

To complete the proof, we combine Equations (B.1), (B.3), and (B.4), to obtain

$$L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}, \boldsymbol{\ell}^t) \geq L_{avg}^t + \frac{\delta(m)}{6}.$$

The best voter in hindsight incurs loss that is at most as high as the average voter. Therefore, the overall regret is

$$Reg_T = \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{w}^t}, \boldsymbol{\ell}^t) - \min_i \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t, \mathbf{e}_i}, \boldsymbol{\ell}^t)$$

$$\geq \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}, \boldsymbol{\ell}^t) - \sum_{t=1}^{T} L^t_{avg}$$
$$\geq T\frac{\delta(m)}{6}.$$

In words, the weighting scheme suffers linear regret. □

B.4 The Stronger Benchmark: Best Weights in Hindsight

In this section, we discuss our results as they apply to the stronger benchmark of competing with the best voter weights in hindsight.

Our goal is to design a weighting scheme that weights the rankings of the voters at each time step, and elects winners with overall expected loss that is almost as small as that of the *best voter weights in hindsight.* We refer to the expected difference between these losses as the expected *regret* with respect to the best weight in hindsight benchmark. That is,

$$\mathbb{E}[Reg_T] \triangleq \mathbb{E}\left[\sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}^t}, \boldsymbol{\ell}^t) - \min_{\mathbf{w}:\|\mathbf{w}\|_1=1} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}}, \boldsymbol{\ell}^t)\right].$$

We wish to formalize the claim, made in Section 3.6, that Theorem 3.5.3 holds under the stronger benchmark. We do this by showing that, indeed, for distributions over unilaterals, the best-weights-in-hindsight benchmark is equivalent to the best voter in hindsight.

Theorem B.4.1. *For any voting rule that is a distribution over unilaterals, there exist deterministic weighting schemes with regret of* $O(\sqrt{T\ln(n)})$ *and* $O(\sqrt{Tn\ln(n)})$ *with respect to the best weight in hindsight benchmark, in the full-information and partial-information settings, respectively.*

Proof. It suffices to show that

$$\min_{\mathbf{w}:\|\mathbf{w}\|_1=1} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{w}}, \boldsymbol{\ell}^t) = \min_{i\in[n]} \sum_{t=1}^{T} L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma}^t,\mathbf{e}_i}, \boldsymbol{\ell}^t), \tag{B.8}$$

as then the theorem follows from Theorem 3.5.3. In turn, to prove Equation (B.8) it is sufficient to show that $L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}, \boldsymbol{\ell})$ is a linear function in $\mathbf{w}$, because any linear function is optimized at an extreme point of the convex set $\{\mathbf{w} \mid \|\mathbf{w}\|_1 = 1\}$.

Let f be a distribution over unilaterals $g_1, \ldots, g_k$ with corresponding probabilities $q_1, \ldots, q_k$. Also, let $h_j : \mathcal{L}(A) \to A$ denote the function corresponding to g_j, for $j \in [k]$. Given a weight vector $\mathbf{w}$ such that $\|\mathbf{w}\|_1 = 1$ and $\boldsymbol{\sigma}$, let $\boldsymbol{\pi} = \boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}$. It holds that

$$f(\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}) = \sum_{j=1}^{k} q_j \sum_{\tau\in\mathcal{L}(A)} \pi_\tau \mathbf{e}_{h_j(\tau)}$$

$$= \sum_{j=1}^{k} q_j \sum_{\tau \in \mathcal{L}(A)} \mathbf{e}_{h_j(\tau)} \sum_{i=1}^{n} w_i \mathbb{1}_{(\sigma_i = \tau)}$$
$$= \sum_{i=1}^{n} w_i \sum_{j=1}^{k} q_j \mathbf{e}_{h_j(\sigma_i)}.$$

Therefore,

$$L_f(\boldsymbol{\pi}_{\boldsymbol{\sigma},\mathbf{w}}, \boldsymbol{\ell}) = \sum_{i=1}^{n} w_i \sum_{j=1}^{k} q_j \left(\mathbf{e}_{h_j(\sigma_i)} \cdot \boldsymbol{\ell} \right)$$
$$= \sum_{i=1}^{n} w_i \sum_{j=1}^{k} q_j \ell_{h_j(\sigma_i)};$$

the right hand side is clearly linear in $\mathbf{w}$. □

Appendix C

Omitted Proofs and Results for Chapter 4

C.1 Proof of Theorem 4.3.1 and Omitted Lemmas

C.1.1 Proof of Lemma 4.3.4

Let f be an anonymous SCC that satisfies monotonicity and neutrality. Let π be an arbitrary anonymous preference profile, and let a, b be two arbitrary alternatives such that $a \triangleright_\pi b$. Now, suppose for the sake of contradiction that $b \in f(\pi)$ but $a \notin f(\pi)$.

Consider an arbitrary ranking σ with $a \succ_\sigma b$. Since $a \triangleright_\pi b$, $\pi(\sigma) \geq \pi(\sigma^{ab})$. In other words, we have an excess weight of $\pi(\sigma) - \pi(\sigma^{ab})$ on σ. For this excess weight of σ, move b upwards and place it just below a. By monotonicity, b still wins and a still loses in this modified profile. We repeat this procedure for every such σ (i.e. for its excess weight, move b upwards, until it is placed below a). In the resulting profile, a still loses. Now, for each of the modified rankings, move a down to where b originally was. By monotonicity, a still loses in the resulting profile π', i.e., $a \notin f(\pi')$.

On the other hand, this procedure is equivalent to shifting the excess weight $\pi(\sigma) - \pi(\sigma^{ab})$ from σ to σ^{ab} (for each σ with $a \succ_\sigma b$). Hence, the profile π' we end up with is such that $\pi'(\sigma) = \pi(\sigma^{ab})$ and $\pi'(\sigma^{ab}) = \pi(\sigma)$, i.e. the new profile is the original profile with a and b swapped. Therefore, by neutrality, it must be the case that $a \in f(\pi')$. This contradicts our conclusion that $a \notin f(\pi')$, thus completing the proof. □

C.1.2 Proof of Theorem 4.3.7

Let f, Π, and A as in the theorem statement. Since Π is SwD-compatible, $\triangleright_\Pi$ is a total preorder on $\mathcal{X}$. In turn, the relation $\triangleright_\Pi$ restricted to A is a total preorder on A. Therefore, there is $a \in A$ such that $a \triangleright_\Pi b$ for all $b \in A$.

Suppose for the sake of contradiction that $a \notin f(\Pi(A))$, and let $b \in A \setminus \{a\}$. Then it holds that $a \triangleright_\Pi b$. In particular, $a \triangleright_{\Pi(A)} b$. But, because f is SwD-efficient and $a \notin f(\Pi(A))$, we have that $b \notin f(\Pi(A))$. This is true for every $b \in A$, leading to $f(\Pi(A)) = \phi$, which contradicts the definition of an SCC. □

C.1.3 Proof of Lemma 4.3.9

Let a and b be two alternatives such that a dominates b in U. In addition, let A be a finite set of alternatives containing a and b, let π denote the anonymous preference profile $\Pi(A)$, and let $m = |A|$. Consider an arbitrary ranking σ such that $a \succ_\sigma b$. Now, let $x_\ell = \sigma^{-1}(\ell)$ denote the alternative in position ℓ of σ, and let $i = \sigma(a)$, $j = \sigma(b)$, i.e.,

$$x_1 \succ_\sigma x_2 \cdots \succ_\sigma x_i (= a) \succ_\sigma \cdots \succ_\sigma x_j (= b) \succ_\sigma \cdots \succ_\sigma x_m.$$

Then,

$$\begin{aligned}\pi(\sigma) &= P(U_{x_1} > U_{x_2} > \cdots > U_{x_i} > \cdots > U_{x_j} > \cdots > U_{x_m}) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{u_1} \cdots \int_{-\infty}^{u_{i-1}} \cdots \int_{-\infty}^{u_{j-1}} \cdots \int_{-\infty}^{u_{m-1}} \\ &\qquad p(u_1, u_2, \cdots, u_i, \cdots u_j, \cdots, u_m) du_m \cdots du_1.\end{aligned}$$

In this integral, because of the limits, we always have $u_i \geq u_j$. Moreover, since $x_i = a$ dominates $x_j = b$ in U, we have

$$\begin{aligned}\pi(\sigma) \geq &\int_{-\infty}^{\infty} \int_{-\infty}^{u_1} \cdots \int_{-\infty}^{u_{i-1}} \cdots \int_{-\infty}^{u_{j-1}} \cdots \int_{-\infty}^{u_{m-1}} \\ &\qquad p(u_1, u_2, \cdots, u_j, \cdots u_i, \cdots, u_m) du_m \cdots du_1.\end{aligned}$$

The right-hand side of this equation is exactly $\pi(\sigma^{ab})$. Hence, we have $\pi(\sigma) \geq \pi(\sigma^{ab})$. It follows that $a \rhd_\pi b$, i.e., $a \rhd_{\Pi(A)} b$. Also, this is true for any finite A containing a and b. We conclude that $a \rhd_\Pi b$. □

C.1.4 Proof of Lemma 4.3.10

We establish the property separately for the TM and PL processes.

TM process. Let a and b be two alternatives such that $\mu_a \geq \mu_b$. Since we are dealing with a TM process, $U_a \sim \mathcal{N}(\mu_a, \frac{1}{2})$ and $U_b \sim \mathcal{N}(\mu_b, \frac{1}{2})$. Let A be any finite set of alternatives containing a and b. Since utilities are sampled independently in a TM process, the difference between the two sides of Equation (4.1) is that the left-hand side has $p_{U_a}(u_1)p_{U_b}(u_2)$, while the right-hand side has $p_{U_a}(u_2)p_{U_b}(u_1)$. It holds that

$$\begin{aligned}&p_{U_a}(u_1)p_{U_b}(u_2) \\ &\quad = \frac{1}{\sqrt{\pi}} \exp\left(-(u_1 - \mu_a)^2\right) \frac{1}{\sqrt{\pi}} \exp\left(-(u_2 - \mu_b)^2\right). \\ &\quad = \frac{1}{\pi} \exp\left(-u_1^2 - \mu_a^2 - u_2^2 - \mu_b^2 + 2u_1\mu_a + 2u_2\mu_b\right).\end{aligned} \tag{C.1}$$

We have $u_1 \geq u_2$ and $\mu_a \geq \mu_b$. Therefore,

$$u_1\mu_a + u_2\mu_b = u_1\mu_b + u_1(\mu_a - \mu_b) + u_2\mu_b$$

$$
\begin{aligned}
&\geq u_1\mu_b + u_2(\mu_a - \mu_b) + u_2\mu_b \\
&= u_1\mu_b + u_2\mu_a
\end{aligned}
$$

Substituting this into Equation (C.1), we obtain

$$
\begin{aligned}
&p_{U_a}(u_1)p_{U_b}(u_2) \\
&\quad\geq \frac{1}{\pi}\exp\left(-u_1^2 - \mu_a^2 - u_2^2 - \mu_b^2 + 2u_1\mu_b + 2u_2\mu_a\right) \\
&\quad= \frac{1}{\pi}\exp\left(-(u_2 - \mu_a)^2 - (u_1 - \mu_b)^2\right) \\
&\quad= p_{U_a}(u_2)p_{U_b}(u_1)
\end{aligned}
$$

It follows that Equation (4.1) holds true. Hence, a dominates b in the corresponding utility process.

To show the other direction, let a and b be such that $\mu_a < \mu_b$. If we choose u_1, u_2 such that $u_1 > u_2$, using a very similar approach as above, we get $p_{U_a}(u_1)p_{U_b}(u_2) < p_{U_a}(u_2)p_{U_b}(u_1)$. And so, a does not dominate b in the corresponding utility process. □

PL process. Let a and b be two alternatives such that $\mu_a \geq \mu_b$. Since we are dealing with a PL process, $U_a \sim \mathcal{G}(\mu_a, \gamma)$ and $U_b \sim \mathcal{G}(\mu_b, \gamma)$. Let A be any finite set of alternatives containing a and b. Since utilities are sampled independently in a PL process, the difference between the two sides of Equation (4.1) is that the left-hand side has $p_{U_a}(u_1)p_{U_b}(u_2)$, while the right-hand side has $p_{U_a}(u_2)p_{U_b}(u_1)$. It holds that

$$
\begin{aligned}
&p_{U_a}(u_1)p_{U_b}(u_2) \\
&\quad= \frac{1}{\gamma}\exp\left(-\frac{u_1 - \mu_a}{\gamma} - e^{-\frac{u_1-\mu_a}{\gamma}}\right)\frac{1}{\gamma}\exp\left(-\frac{u_2 - \mu_b}{\gamma} - e^{-\frac{u_2-\mu_b}{\gamma}}\right) \\
&\quad= \frac{1}{\gamma^2}\exp\left(-\frac{u_1 - \mu_a}{\gamma} - e^{-\frac{u_1-\mu_a}{\gamma}} - \frac{u_2 - \mu_b}{\gamma} - e^{-\frac{u_2-\mu_b}{\gamma}}\right) \\
&\quad= \frac{1}{\gamma^2}\exp\left(-\frac{u_1 - \mu_a + u_2 - \mu_b}{\gamma} - \left(e^{-\frac{u_1}{\gamma}}e^{\frac{\mu_a}{\gamma}} + e^{-\frac{u_2}{\gamma}}e^{\frac{\mu_b}{\gamma}}\right)\right).
\end{aligned}
\tag{C.2}
$$

We also know that $e^{-\frac{u_2}{\gamma}} \geq e^{-\frac{u_1}{\gamma}}$ and $e^{\frac{\mu_a}{\gamma}} \geq e^{\frac{\mu_b}{\gamma}}$. Similar to the proof for the TM process, we have

$$
e^{-\frac{u_2}{\gamma}}e^{\frac{\mu_a}{\gamma}} + e^{-\frac{u_1}{\gamma}}e^{\frac{\mu_b}{\gamma}} \geq e^{-\frac{u_1}{\gamma}}e^{\frac{\mu_a}{\gamma}} + e^{-\frac{u_2}{\gamma}}e^{\frac{\mu_b}{\gamma}}.
$$

Substituting this into Equation (C.2), we obtain

$$
\begin{aligned}
&p_{U_a}(u_1)p_{U_b}(u_2) \\
&\quad\geq \frac{1}{\gamma^2}\exp\left(-\frac{u_1 - \mu_a + u_2 - \mu_b}{\gamma} - \left(e^{-\frac{u_2}{\gamma}}e^{\frac{\mu_a}{\gamma}} + e^{-\frac{u_1}{\gamma}}e^{\frac{\mu_b}{\gamma}}\right)\right) \\
&\quad= \frac{1}{\gamma}\exp\left(-\frac{u_2 - \mu_a}{\gamma} - e^{-\frac{u_2-\mu_a}{\gamma}}\right)\frac{1}{\gamma}\exp\left(-\frac{u_1 - \mu_b}{\gamma} - e^{-\frac{u_1-\mu_b}{\gamma}}\right) \\
&\quad= p_{U_a}(u_2)p_{U_b}(u_1)
\end{aligned}
$$

It follows that Equation (4.1) holds true. Hence, a dominates b in the corresponding utility process.

To show the other direction, let a and b be such that $\mu_a < \mu_b$. If we choose u_1, u_2 such that $u_1 > u_2$, using a very similar approach as above, we get $p_{U_a}(u_1)p_{U_b}(u_2) < p_{U_a}(u_2)p_{U_b}(u_1)$. And so, a does not dominate b in the corresponding utility process. □

C.1.5 Proof of Theorem 4.3.1

By Lemma 4.3.4, the anonymous SCC f is SwD-efficient. Lemmas 4.3.9 and 4.3.10 directly imply that when Π is the TM or PL process, $\rhd_\Pi$ is indeed a total preorder. In particular, $a \rhd_\Pi b$ if $\mu_a \geq \mu_b$. So, an alternative a in A with maximum mode utility satisfies $a \rhd_\Pi b$ for all $b \in A$. By Theorem 4.3.7, if $a \in A$ is such that $a \rhd_\Pi b$ for all $b \in A$, then $a \in f(\Pi(A))$; the statement of the theorem follows. □

C.2 More on Stability and Proof of Theorem 4.3.12

Before proving Theorem 4.3.12, we examine some examples that illustrate stability (or the lack thereof).

Example C.2.1. Let f be the Borda count SCC, and let the set of alternatives be $\mathcal{X} = \{u, v, w, x, y\}$. Also, let Π be a consistent permutation process, which, given all the alternatives, gives a uniform distribution on the two rankings $(x \succ u \succ v \succ y \succ w)$ and $(y \succ w \succ x \succ u \succ v)$. The outcome of applying f on this profile is $\{x\}$ (since x has the strictly highest Borda score). But, the outcome of applying f on the profile $\Pi(\{w, x, y\})$ is $\{y\}$ (since y now has the strictly highest Borda score). Hence, $f(\Pi(\{u, v, w, x, y\})) \cap \{w, x, y\} \neq f(\Pi(w, x, y))$, even though the left-hand side is nonempty. We conclude that the tuple (Π, f) does not satisfy stability.

For the next example (and the statement of Theorem 4.3.12), we need to define the *Copeland* SCC. For an anonymous preference profile π over A, we say that $a \in A$ beats $b \in A$ in a pairwise election if

$$\sum_{\sigma \in \mathcal{S}_A : a \succ_\sigma b} \pi(\sigma) > \frac{1}{2}.$$

The *Copeland score* of an alternative is the number of other alternatives it beats in pairwise elections; the Copeland SCC selects all alternatives that maximize the Copeland score.

Example C.2.2. Consider the permutation process of Example C.2.1, and let f be the Copeland SCC. Once again, it holds that $f(\Pi(u, v, w, x, y)) = \{x\}$ and $f(\Pi(w, x, y)) = \{y\}$. Hence the pair (Π, f) is not stable.

Now, in the spirit of Theorem 4.3.7, let us see whether the pair (Π, f) satisfies stability when f is an SwD-efficient anonymous SCC, and Π is an SwD-compatible permutation process. Example C.2.3 constructs such a Π that is not stable with respect to the plurality SCC (even though plurality is SwD-efficient).

Example C.2.3. Let f be the plurality SCC and the set of alternatives be $\mathcal{X} = \{a, b, c\}$. Also, let Π be the consistent permutation process, which given all alternatives, gives the

following profile: 0.35 weight on $(a \succ b \succ c)$, 0.35 weight on $(b \succ a \succ c)$, 0.1 weight on $(c \succ a \succ b)$, 0.1 weight on $(a \succ c \succ b)$ and 0.1 weight on $(b \succ c \succ a)$. All the swap-dominance relations in this permutation process are: $a \rhd_\Pi b$, $b \rhd_\Pi c$ and $a \rhd_\Pi c$. Hence, $\rhd_\Pi$ is a total preorder on $\mathcal{X}$, and Π is SwD-compatible.
Now, for this permutation process Π and the plurality SCC f, we have: $f(\Pi(\{a,b,c\})) = \{a,b\}$ and $f(\Pi(\{a,b\})) = \{a\}$. Therefore, (Π, f) is not stable.

This happens because Plurality is not *strongly* SwD-efficient, as defined below (Example C.2.3 even shows why plurality violates this property).

Definition C.2.4. An anonymous SCC f is said to be *strongly SwD-efficient* if for every anonymous preference profile π over A, and any two alternatives $a, b \in A$ such that $a \rhd_\pi b$,

1. If $b \not\rhd_\pi a$, then $b \notin f(\pi)$.
2. If $b \rhd_\pi a$, then $b \in f(\pi) \Leftrightarrow a \in f(\pi)$.

It is clear that any strongly SwD-efficient SCC is also SwD-efficient.

Lemma C.2.5. *The Borda count and Copeland SCCs are strongly SwD-efficient.*

Proof. Let π be an arbitrary anonymous preference profile over alternatives A, and let $a, b \in A$ such that $a \rhd_\pi b$. This means that for all $\sigma \in \mathcal{S}_A$ with $a \succ_\sigma b$, we have $\pi(\sigma) \geq \pi(\sigma^{ab})$. We will examine the two conditions (of Definition C.2.4) separately.

Case 1: $b \not\rhd_\pi a$. This means that there exists a ranking $\sigma_* \in \mathcal{S}_A$ with $b \succ_{\sigma_*} a$ such that $\pi(\sigma_*) < \pi(\sigma_*^{ab})$. Below we analyze each of the SCCs mentioned in the theorem.

Borda count. $\mathcal{S}_A$ can be partitioned into pairs of the form (σ, σ^{ab}), where σ is such that $a \succ_\sigma b$. We reason about how each pair contributes to the Borda scores of a and b. Consider an arbitrary pair (σ, σ^{ab}) with $a \succ_\sigma b$. The score contributed by σ to a is $(m - \sigma(a))\pi(\sigma)$, and the score contributed to b is $(m - \sigma(b))\pi(\sigma)$. That is, it gives an excess score of $(\sigma(b) - \sigma(a))\pi(\sigma)$ to a. Similarly, the score of a contributed by σ^{ab} is $(m - \sigma^{ab}(a))\pi(\sigma^{ab}) = (m - \sigma(b))\pi(\sigma^{ab})$, and the score contributed to b is $(m - \sigma(a))\pi(\sigma^{ab})$. So, b gets an excess score of $(\sigma(b) - \sigma(a))\pi(\sigma^{ab})$ from σ^{ab}. Combining these observations, the pair (σ, σ^{ab}) gives a an excess score of $(\sigma(b) - \sigma(a))(\pi(\sigma) - \pi(\sigma^{ab}))$, which is at least 0. Since this is true for every pair (σ, σ^{ab}), a has Borda score that is at least as high as that of b. Furthermore, the pair $(\sigma_*^{ab}, \sigma_*)$ is such that $\pi(\sigma_*^{ab}) - \pi(\sigma_*) > 0$, so, this pair gives a an excess score that is strictly positive. We conclude that a has strictly higher Borda score than b, hence b is not selected by Borda count.

Copeland. Let $c \in A \setminus \{a, b\}$. In a pairwise election between b and c, the total weight of rankings that place b over c is

$$\sum_{\sigma \in \mathcal{S}_A : b \succ_\sigma c} \pi(\sigma) = \sum_{\sigma \in \mathcal{S}_A : (b \succ_\sigma c) \wedge (a \succ_\sigma c)} \pi(\sigma) + \sum_{\sigma \in \mathcal{S}_A : (b \succ_\sigma c) \wedge (c \succ_\sigma a)} \pi(\sigma).$$

For the rankings in the second summation (on the right-hand side), we have $b \succ_\sigma a$ by transitivity. Hence, $\pi(\sigma) \leq \pi(\sigma^{ab})$ for such rankings. Therefore,

$$\sum_{\sigma \in \mathcal{S}_A : b \succ_\sigma c} \pi(\sigma) \leq \sum_{\sigma \in \mathcal{S}_A : (b \succ_\sigma c) \wedge (a \succ_\sigma c)} \pi(\sigma) + \sum_{\sigma \in \mathcal{S}_A : (b \succ_\sigma c) \wedge (c \succ_\sigma a)} \pi(\sigma^{ab})$$

$$= \sum_{\sigma \in \mathcal{S}_A : (b \succ_\sigma c) \wedge (a \succ_\sigma c)} \pi(\sigma) + \sum_{\sigma' \in \mathcal{S}_A : (a \succ_{\sigma'} c) \wedge (c \succ_{\sigma'} b)} \pi(\sigma')$$
$$= \sum_{\sigma \in \mathcal{S}_A : a \succ_\sigma c} \pi(\sigma).$$

In summary, we have

$$\sum_{\sigma \in \mathcal{S}_A : b \succ_\sigma c} \pi(\sigma) \leq \sum_{\sigma \in \mathcal{S}_A : a \succ_\sigma c} \pi(\sigma).$$

Hence, if b beats c in a pairwise competition, then so does a. Therefore, the Copeland score of a (due to all alternatives other than a and b) is at least as high as that of b. Further, in a pairwise competition between a and b, the weight of rankings that position a above b is $\sum_{\sigma \in \mathcal{S}_A : a \succ_\sigma b} \pi(\sigma)$ and the weight of those that prefer b over a is $\sum_{\sigma \in \mathcal{S}_A : b \succ_\sigma a} \pi(\sigma)$. But, because $\pi(\sigma) \geq \pi(\sigma^{ab})$ for any σ with $a \succ_\sigma b$, and $\pi(\sigma^{ab}_*) > \pi(\sigma_*)$, a beats b. Therefore, a has a strictly higher Copeland score than b, and b is not selected by Copeland.

Case 2: $b \rhd_\pi a$. In this case, $a \rhd_\pi b$ and $b \rhd_\pi a$. This means that for all $\sigma \in \mathcal{S}_A$, we have $\pi(\sigma) = \pi(\sigma^{ab})$. In other words, $\tau(\pi) = \pi$, where τ is the permutation that swaps a and b. Both Borda count and Copeland are neutral SCCs. So, we have $\tau(f(\pi)) = f(\tau(\pi))$, which is in turn equal to $f(\pi)$. Hence, a is selected if and only if b is selected.

We conclude that both conditions of Definition C.2.4 are satisfied by Borda count and Copeland. □

Lemma C.2.6. *Let* Π *be a consistent permutation process that is SwD-compatible. Then, for any finite subset of alternatives* $A \subseteq \mathcal{X}$, $\left(\rhd_{\Pi(A)}\right) = \left(\rhd_\Pi|_A\right)$.

In words, as long as Π is consistent and SwD-compatible, marginalizing out some alternatives from a profile does not remove or add any swap-dominance relations.

Proof of Lemma C.2.6. We first show that for any $B \subseteq A \subseteq \mathcal{X}$, $\left(\rhd_{\Pi(A)}|_B\right) = \left(\rhd_{\Pi(B)}\right)$.

Let $a, b \in B$ such that $a \rhd_{\Pi(A)} b$. Now, let $\sigma \in \mathcal{S}_B$ be an arbitrary ranking such that $a \succ_\sigma b$. Also, let π_B denote $\Pi(B)$ and π_A denote $\Pi(A)$. Then, since Π is consistent,

$$\pi_B(\sigma) = \sum_{\sigma_2 \in \mathcal{S}_A : \sigma_2|_B = \sigma} \pi_A(\sigma_2).$$

Now, for $\sigma_2 \in \mathcal{S}_A$ such that $\sigma_2|_B = \sigma$, we have $a \succ_{\sigma_2} b$ and therefore $\pi_A(\sigma_2) \geq \pi_A(\sigma_2^{ab})$ (because $a \rhd_{\Pi(A)} b$). It follows that

$$\pi_B(\sigma) = \sum_{\sigma_2 \in \mathcal{S}_A : \sigma_2|_B = \sigma} \pi_A(\sigma_2) \geq \sum_{\sigma_2 \in \mathcal{S}_A : \sigma_2|_B = \sigma} \pi_A(\sigma_2^{ab})$$
$$= \sum_{\sigma_2' \in \mathcal{S}_A : \sigma_2'|_B = \sigma^{ab}} \pi_A(\sigma_2')$$
$$= \pi_B(\sigma^{ab}).$$

Therefore, $a \rhd_{\Pi(B)} b$, that is, $\left(\rhd_{\Pi(A)}|_B\right) \subseteq \left(\rhd_{\Pi(B)}\right)$.

Next we show that $\left(\rhd_{\Pi(B)}\right) \subseteq \left(\rhd_{\Pi(A)}|_B\right)$. Let $a, b \in B$ such that $a \rhd_{\Pi(B)} b$. Suppose for the sake of contradiction that $a \not\rhd_{\Pi(A)} b$. This implies that $a \not\rhd_{\Pi} b$. However, $\rhd_{\Pi}$ is a total preorder because Π is SwD-compatible (by definition). It follows that $b \rhd_{\Pi} a$, and, in particular, $b \rhd_{\Pi(A)} a$ and $b \rhd_{\Pi(B)} a$.

As before, let π_A denote $\Pi(A)$ and π_B denote $\Pi(B)$. Because $a \not\rhd_{\Pi(A)} b$, there exists $\sigma_* \in \mathcal{S}_A$ with $a \succ_{\sigma_*} b$ such that $\pi_A(\sigma_*) < \pi_A(\sigma_*^{ab})$. Moreover, because $a \rhd_{\Pi(B)} b$ and $b \rhd_{\Pi(B)} a$, it holds that $\pi_B(\sigma_*|_B) = \pi_B\left((\sigma_*|_B)^{ab}\right)$. The consistency of Π then implies that

$$\sum_{\sigma_1 \in \mathcal{S}_A : \sigma_1|_B = \sigma_*|_B} \pi_A(\sigma_1) = \sum_{\sigma_2 \in \mathcal{S}_A : \sigma_2|_B = (\sigma_*|_B)^{ab}} \pi_A(\sigma_2). \tag{C.3}$$

Note $\sigma_1 = \sigma_*$ is a ranking that appears on the left-hand side of Equation (C.3), and $\sigma_2 = \sigma_*^{ab}$ is a ranking that appears on the right-hand side. Furthermore, we know that $\pi_A(\sigma_*) < \pi_A(\sigma_*^{ab})$. It follows that there exists $\sigma' \in \mathcal{S}_A$ with $\sigma'|_B = \sigma_*|_B$ such that $\pi_A(\sigma') > \pi_A\left((\sigma')^{ab}\right)$. Also, since $\sigma'|_B = \sigma_*|_B$, it holds that $a \succ_{\sigma'} b$. We conclude that it cannot be the case that $b \rhd_{\Pi(A)} a$, leading to a contradiction. Therefore, if $a \rhd_{\Pi(B)} b$, then $a \rhd_{\Pi(A)} b$, i.e., $\left(\rhd_{\Pi(B)}\right) \subseteq \left(\rhd_{\Pi(A)}|_B\right)$.

We next prove the lemma itself, i.e., that $\left(\rhd_{\Pi(A)}\right) = \left(\rhd_{\Pi}|_A\right)$. Firstly, for $a, b \in A$, if $a \rhd_{\Pi} b$, then $a \rhd_{\Pi(A)} b$ by definition. So, we easily get $\left(\rhd_{\Pi}|_A\right) \subseteq \left(\rhd_{\Pi(A)}\right)$.

In the other direction, let $a, b \in A$ such that $a \rhd_{\Pi(A)} b$. Let C be an arbitrary set of alternatives containing a and b. From what we have shown above, we have $\left(\rhd_{\Pi(A)}|_{\{a,b\}}\right) = \left(\rhd_{\Pi(\{a,b\})}\right)$. Also, $\left(\rhd_{\Pi(C)}|_{\{a,b\}}\right) = \left(\rhd_{\Pi(\{a,b\})}\right)$. This gives us $\left(\rhd_{\Pi(A)}|_{\{a,b\}}\right) = \left(\rhd_{\Pi(C)}|_{\{a,b\}}\right)$. Hence, $a \rhd_{\Pi(C)} b$, and this is true for every such subset C. We conclude that $a \rhd_{\Pi} b$, that is, $\left(\rhd_{\Pi(A)}\right) \subseteq \left(\rhd_{\Pi}|_A\right)$. □

Lemma C.2.7. *Let f be a strongly SwD-efficient anonymous SCC, and let Π be a consistent permutation process that is SwD-compatible. Then for any finite subset of alternatives A, $f(\Pi(A)) = \{a \in A : a \rhd_{\Pi} b \text{ for all } b \in A\}$.*

Proof. Let A be an arbitrary finite subset of alternatives. Since strong SwD-efficiency implies SwD-efficiency, Theorem 4.3.7 gives us

$$f(\Pi(A)) \supseteq \{a \in A : a \rhd_{\Pi} b \text{ for all } b \in A\}.$$

In the other direction, let $a \in f(\Pi(A))$. Suppose for the sake of contradiction that there exists $b \in A$ such that $a \not\rhd_{\Pi} b$. Since $\rhd_{\Pi}$ is a total preorder, it follows that $b \rhd_{\Pi} a$. By Lemma C.2.6, it holds that $\left(\rhd_{\Pi(A)}\right) = \left(\rhd_{\Pi}|_A\right)$, and therefore $a \not\rhd_{\Pi(A)} b$ and $b \rhd_{\Pi(A)} a$. But, since f is strongly SwD-efficient, it follows that $a \notin f(\Pi(A))$, which contradicts our assumption. Hence,

$$f(\Pi(A)) \subseteq \{a \in A : a \rhd_{\Pi} b \text{ for all } b \in A\},$$

and we have the desired result. □

Theorem C.2.8. *Let Π be a consistent permutation process that is SwD-compatible, and let f be a strongly SwD-efficient anonymous SCC. Then the pair (Π, f) is stable.*

Proof. Consider an arbitrary subset of alternatives A, and let $B \subseteq A$. By Lemma C.2.7, $f(\Pi(A)) = \{a \in A : a \triangleright_\Pi b \text{ for all } b \in A\}$, and similarly for B. Suppose $f(\Pi(A)) \cap B \neq \phi$, and let $a \in f(\Pi(A)) \cap B$, i.e. $a \in f(\Pi(A))$ and $a \in B$. This means that $a \triangleright_\Pi b$ for all $b \in A$, and, therefore $a \triangleright_\Pi b$ for all $b \in B$. We conclude that $a \in f(\Pi(B))$, and hence $f(\Pi(A)) \cap B \subseteq f(\Pi(B))$.

In the other direction, let $a \in f(\Pi(B))$. This means that $a \triangleright_\Pi b$ for all $b \in B$. Suppose for the sake of contradiction that $a \notin f(\Pi(A))$. This means that there exists $c \in A$ such that $a \not\triangleright_\Pi c$. We assumed $f(\Pi(A)) \cap B \neq \phi$, so let $d \in f(\Pi(A)) \cap B$. Then, $d \triangleright_\Pi c$. In summary, we have $d \triangleright_\Pi c$ and $a \not\triangleright_\Pi c$, which together imply that $a \not\triangleright_\Pi d$ (otherwise, it would violate transitivity). But $d \in B$, leading to $a \notin f(\Pi(B))$, which contradicts the assumption. Therefore, indeed $a \in f(\Pi(A))$, and it holds that $f(\Pi(B)) \subseteq f(\Pi(A)) \cap B$, as long as $f(\Pi(A)) \cap B \neq \phi$. □

We are now ready to prove Theorem 4.3.12.

Proof of Theorem 4.3.12. From Lemma C.2.5, Borda count and Copeland are strongly SwD-efficient. Lemmas 4.3.9 and 4.3.10 imply that when Π is the TM or PL process, $\triangleright_\Pi$ is a total preorder. In particular, $a \triangleright_\Pi b$ if $\mu_a \geq \mu_b$. Hence, Π is SwD-compatible. Therefore, by Theorem C.2.8, the pair (Π, f) is stable. □

C.3 Proof of Proposition 4.4.1

Let $\bar{\boldsymbol{\beta}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\beta}_i$. We know that $U_x^{\boldsymbol{\beta}}$ denotes the utility of x under the TM process with parameter $\boldsymbol{\beta}$. So, $U_x^{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}^\intercal x, \frac{1}{2})$. Let its density be given by $q_{x,\boldsymbol{\beta}}(\cdot)$. Also, $U_x^{\boldsymbol{\beta}_i} \sim \mathcal{N}(\boldsymbol{\beta}_i^\intercal x, \frac{1}{2})$. Hence, $\frac{1}{N}\sum_{i=1}^N U_x^{\boldsymbol{\beta}_i} \sim \mathcal{N}(\bar{\boldsymbol{\beta}}^\intercal x, \frac{1}{2N})$. Let its density function be denoted by $p_x(\cdot)$. Then

$$KL(p_x \| q_{x,\boldsymbol{\beta}}) = \int p_x(t) \log p_x(t) dt - \int p_x(t) \log q_{x,\boldsymbol{\beta}}(t) dt.$$

Since the first term does not depend on $\boldsymbol{\beta}$, let us examine the second term:

$$\begin{aligned}
-\int p_x(t) \log q_{x,\boldsymbol{\beta}}(t) dt &= -\int p_x(t) \log\left(\frac{1}{\sqrt{\pi}} \exp\left(-(t - \boldsymbol{\beta}^\intercal x)^2\right)\right) dt \\
&= -\int p_x(t) \left[-\frac{1}{2}\log(\pi) - (t - \boldsymbol{\beta}^\intercal x)^2\right] dt \\
&= \frac{1}{2}\log(\pi)\left(\int p_x(t) dt\right) + \int p_x(t)\left(t^2 + (\boldsymbol{\beta}^\intercal x)^2 - 2t\boldsymbol{\beta}^\intercal x\right) dt \\
&= \frac{1}{2}\log(\pi) + \left(\int t^2 p_x(t) dt + (\boldsymbol{\beta}^\intercal x)^2 \int p_x(t) dt - 2\boldsymbol{\beta}^\intercal x \int t p_x(t) dt\right) \\
&= \frac{1}{2}\log(\pi) + \left(\left(\frac{1}{2N} + (\bar{\boldsymbol{\beta}}^\intercal x)^2\right) + (\boldsymbol{\beta}^\intercal x)^2 - 2\boldsymbol{\beta}^\intercal x(\bar{\boldsymbol{\beta}}^\intercal x)\right) \\
&= \frac{1}{2}\log(\pi) + \frac{1}{2N} + \left(\bar{\boldsymbol{\beta}}^\intercal x - \boldsymbol{\beta}^\intercal x\right)^2.
\end{aligned}$$

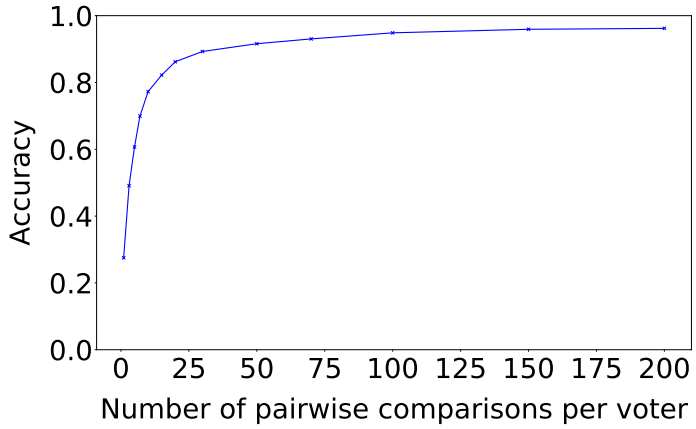


Figure C.1: Accuracy of Step II with number of voters $N = 40$ (synthetic data)

This term is minimized at $\boldsymbol{\beta} = \bar{\boldsymbol{\beta}}$ for any x, and therefore $KL(\frac{1}{N}\sum_{i=1}^{N} U_x^{\boldsymbol{\beta}_i} \| U_x^{\boldsymbol{\beta}})$ is minimized at that value as well. □

C.4 Robustness of the Empirical Results

In Section 4.5.1, we presented experiments using synthetic data, with the following parameters: each instance has 5 alternatives, the number of features is $d = 10$, and, in Step II, we let number of voters be $N = 20$. In this appendix, to demonstrate the robustness of both steps, we show experimental results for different values of these parameters (keeping everything else fixed).

C.4.1 Number of Voters in Step II

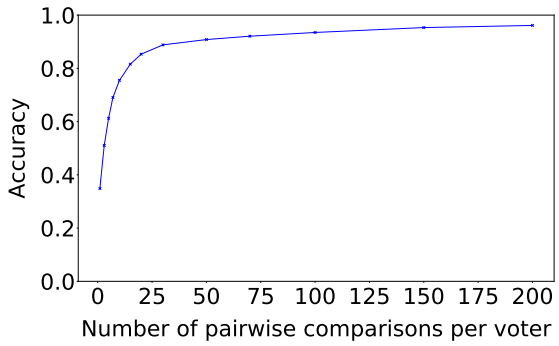
To show robustness with respect to the number of voters N in Step II, we run the Step II experiments with 40 (instead of $N = 20$). The results are shown in Figure C.1.

As before, we observe that the accuracy quickly increases as the number of pairwise comparisons increases, and with just 30 pairwise comparisons we achieve an accuracy of 89.3%. With 100 pairwise comparisons, the accuracy is 94.9%.

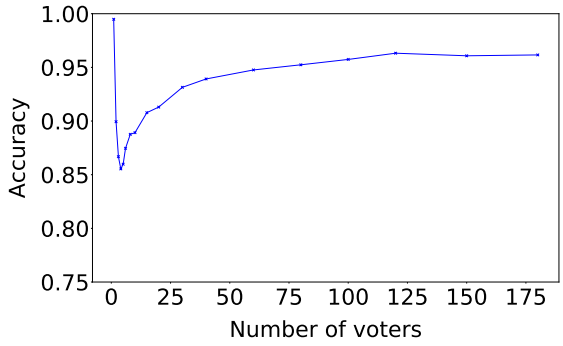
C.4.2 Number of Alternatives

To show robustness with respect to the number of alternatives, we run experiments with $|A| = 3$ (instead of $|A| = 5$). The results are shown in Figure C.2.

Similarly to Section 4.5.1, for Step II, we observe that the accuracy quickly increases as the number of pairwise comparisons increases, and with just 30 pairwise comparisons we achieve an accuracy of 88.8%. With 100 pairwise comparisons, the accuracy is 93.5%. For Step III, we observe that the accuracy increases to 96.2% as the number of voters increases.

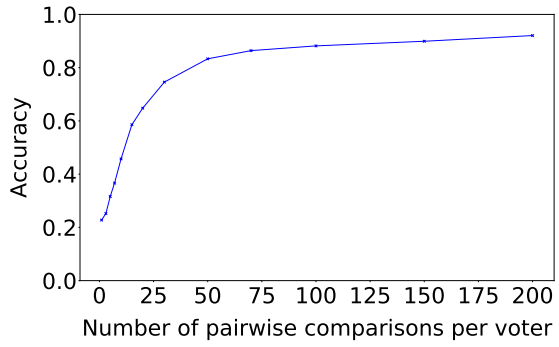


(a) Accuracy of Step II

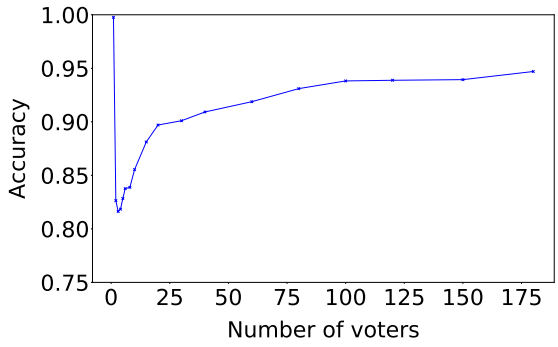


(b) Accuracy of Step III

Figure C.2: Results with 3 alternatives per instance (synthetic data)



(a) Accuracy of Step II



(b) Accuracy of Step III

Figure C.3: Results with number of features $d = 20$ (synthetic data)

C.4.3 Number of Features

To show robustness with respect to the number of features d, we run experiments with $d = 20$ (instead of $d = 10$). The results are shown in Figure C.3.

Again, for Step II, we observe that the accuracy quickly increases (though slower than in Section 4.5.1, because of higher dimension) as the number of pairwise comparisons increases. With just 30 pairwise comparisons we achieve an accuracy of 74.6%, and with 100 pairwise comparisons, the accuracy is 88.2%. For Step III, we observe that the accuracy increases to 94.7% as the number of voters increases.

Appendix D

Omitted Proofs and Results for Chapter 5

D.1 Proof Of Theorem 5.3.1

Recall that the proof of our main result, Theorem 5.3.1, includes four lemmas. Here we prove the three lemmas whose proofs were omitted from the main text.

D.1.1 Proof of Lemma 5.3.3

Consider $L(p,1)$ aggregation with an arbitrary $p \in (1,\infty)$. We show that efficiency is violated using the following construction. There are 2 papers, 3 reviewers and each reviewer reviews both papers. Assume that the papers have objective criteria scores $\mathbf{x}_1$ and $\mathbf{x}_2$, and that neither of these scores is pointwise greater than or equal to the other. Let the overall recommendations by the reviewers for the papers be defined by the matrix

$$\mathbf{y} = \begin{bmatrix} z & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$

where z is a constant strictly bigger than 1 and y_{ia} denotes the overall recommendation by reviewer i to paper a. Observe that paper 1 dominates paper 2. But, we will show that there exists a value $z > 1$ such that the aggregate score of paper 1 is strictly smaller than the aggregate score of paper 2.

Let f_i denote the value of function f on paper i, i.e. $f_i := f(\mathbf{x}_i)$. And let $\widehat{f}_i(z)$ denote the aggregate score of paper i; observe that we write it as a function of z because the aggregate score of each paper would depend on the chosen score z. Since we are minimizing $L(p,1)$ loss, the aggregate function satisfies:

$$(\widehat{f}_1(z), \widehat{f}_2(z)) \in \operatorname*{argmin}_{(f_1,f_2)\in\mathbb{R}^2} \left\{ \left\|(z,0) - (f_1,f_2)\right\|_p + \left\|(0,1) - (f_1,f_2)\right\|_p + \left\|(f_1,f_2)\right\|_p \right\}. \quad \text{(D.1)}$$

We do not have any monotonicity constraints in (D.1) because the two papers have incomparable criteria scores. For simplicity, let $\mathbf{f} := (f_1, f_2)$, $\widehat{\mathbf{f}}(z) := (\widehat{f}_1(z), \widehat{f}_2(z))$, and denote

the objective function in Equation (D.1) by $G_z(\mathbf{f})$. That is,

$$G_z(f_1, f_2) = \Big[|z - f_1|^p + |f_2|^p\Big]^{\frac{1}{p}} + \Big[|f_1|^p + |1 - f_2|^p\Big]^{\frac{1}{p}} + \Big[|f_1|^p + |f_2|^p\Big]^{\frac{1}{p}}. \tag{D.2}$$

For the overall proof to be easier to follow, proofs of all claims are given at the end of this proof. Also, just to re-emphasize, the whole proof assumes $z > 1$.

Claim D.1.1. *G_z is a strictly convex objective function.*

Claim D.1.1 states that G_z is strictly convex, implying that it has a unique minimizer $\widehat{\mathbf{f}}(z)$. Hence, there is no need to consider tie-breaking.

Claim D.1.2. *$\widehat{f}_1(z)$ and $\widehat{f}_2(z)$ are bounded. In particular, $\widehat{f}_1(z) \in [0, 1]$ and $\widehat{f}_2(z) \in [0, 1]$.*

Claim D.1.2 states that the aggregate score of both papers lies in the interval $[0, 1]$ irrespective of the value of z. This allow us to restrict ourselves to the region $[0, 1]^2$ when computing the minimizer of (D.2). Hence, for the rest of the proof, we only consider the space $[0, 1]^2$. In this region, the optimization problem (D.1) can be rewritten as

$$(\widehat{f}_1(z), \widehat{f}_2(z)) = \operatorname*{argmin}_{f_1 \in [0,1], f_2 \in [0,1]} \left\{ \Big[\big(z - f_1\big)^p + f_2^p\Big]^{\frac{1}{p}} + \Big[f_1^p + \big(1 - f_2\big)^p\Big]^{\frac{1}{p}} + \Big[f_1^p + f_2^p\Big]^{\frac{1}{p}} \right\}.$$

To start off, we analyze the objective function as we take the limit of z going to infinity. Later, we show that the observed property holds even for a sufficiently large finite z.

For the limit to exist, redefine the objective function as $H_z(f_1, f_2) = G_z(f_1, f_2) - G_z(0, 0)$, i.e.,

$$H_z(f_1, f_2) = \Big[\big(z - f_1\big)^p + f_2^p\Big]^{\frac{1}{p}} - z + \Big[f_1^p + \big(1 - f_2\big)^p\Big]^{\frac{1}{p}} + \Big[f_1^p + f_2^p\Big]^{\frac{1}{p}} - 1. \tag{D.3}$$

For any value of z, the function H_z has the same minimizer as G_z, that is,

$$(\widehat{f}_1(z), \widehat{f}_2(z)) = \operatorname*{argmin}_{f_1 \in [0,1], f_2 \in [0,1]} H_z(f_1, f_2).$$

Claim D.1.3. *For any (fixed) $f_1 \in [0, 1], f_2 \in [0, 1]$,*

$$\lim_{z \to \infty} H_z(f_1, f_2) = H^\star(f_1, f_2),$$

where

$$H^\star(f_1, f_2) = -f_1 + \Big[f_1^p + \big(1 - f_2\big)^p\Big]^{\frac{1}{p}} + \Big[f_1^p + f_2^p\Big]^{\frac{1}{p}} - 1. \tag{D.4}$$

The proof proceeds by analyzing some important properties of the limiting function $H^\star$.

Claim D.1.4. *The function $H^\star(\mathbf{f})$ is convex in $\mathbf{f} \in [0, 1]^2$. Moreover, the function $H^\star(\mathbf{f})$ is strictly convex for $f_1 \in (0, 1]$ and $f_2 \in [0, 1]$.*

Claim D.1.5. *$H^\star$ is minimized at $\widehat{\mathbf{v}} = (\widehat{v}_1, \widehat{v}_2)$, where*

$$\widehat{v}_1 = \frac{1}{2}\left[\frac{1}{(2^{\frac{p}{p-1}} - 1)}\right]^{\frac{1}{p}}, \qquad \widehat{v}_2 = \frac{1}{2}. \tag{D.5}$$

Claim D.1.6. $\widehat{v}_1 < \widehat{v}_2$.

Observe that Claim D.1.6 is the desired result, but for the limiting objective function $H^\star$. The remainder of the proof proceeds to show that this result holds even for the objective function H_z, when the score z is large enough. Define $\Delta = \widehat{v}_2 - \widehat{v}_1 > 0$. We first show that (i) there exists $z > 1$ such that $\|\widehat{\mathbf{f}}(z) - \widehat{\mathbf{v}}\|_2 < \frac{\Delta}{4}$, and then (ii) show that in this case, we have $\widehat{f}_1(z) < \widehat{f}_2(z)$.

To prove part (i), we first analyze how functions H_z and $H^\star$ relate to each other. Using Claim D.1.3, for any fixed f_1, f_2, by definition of the limit, for any $\epsilon > 0$, there exists z_ϵ (which could be a function of f_1, f_2) such that, for all $z > z_\epsilon$, we have

$$|H_z(f_1, f_2) - H^\star(f_1, f_2)| < \epsilon. \tag{D.6}$$

For a given f_1, f_2, denote the corresponding value of z_ϵ by $z_\epsilon(f_1, f_2)$. And, let $\mathcal{Z}_\epsilon(f_1, f_2)$ denote the set of all values of $z > 1$ for which Equation (D.6) holds for (f_1, f_2).

Claim D.1.7. $\mathcal{Z}_\epsilon(1,1) \subset \mathcal{Z}_\epsilon(f_1, f_2)$ *for every* $(f_1, f_2) \in [0,1]^2$.

Claim D.1.7 says that if Equation (D.6) holds for a particular value of z for $f_1 = f_2 = 1$, then for the same value of z it holds for every other value of $(f_1, f_2) \in [0,1]^2$ as well. So, define

$$\widetilde{z}_\epsilon := z_\epsilon(1,1) + 1. \tag{D.7}$$

By definition, $\widetilde{z}_\epsilon \in \mathcal{Z}_\epsilon(1,1)$. And by Claim D.1.7, $\widetilde{z}_\epsilon \in \mathcal{Z}_\epsilon(f_1, f_2)$ for every $(f_1, f_2) \in [0,1]^2$. So, set $z = \widetilde{z}_\epsilon$. Then, Equation (D.6) holds for all $(f_1, f_2) \in [0,1]^2$ simultaneously. In other words, for all $(f_1, f_2) \in [0,1]^2$, we simultaneously have

$$H^\star(f_1, f_2) - \epsilon < H_z(f_1, f_2) < H^\star(f_1, f_2) + \epsilon, \tag{D.8}$$

i.e. H_z is in an ϵ-band around $H^\star$ throughout this region. And observe that this band gets smaller as ϵ is decreased (which is achieved at a larger value of z).

To bound the distance between $\widehat{\mathbf{v}}$, the minimizer of $H^\star$, and $\widehat{\mathbf{f}}(z)$, the minimizer of H_z, we bound the distance between the objective function values at these points.

Claim D.1.8. $H^\star(\widehat{\mathbf{f}}(z)) < H^\star(\widehat{\mathbf{v}}) + 2\epsilon$.

Although $\widehat{\mathbf{f}}(z)$ does not minimize $H^\star$, Claim D.1.8 says that the objective value at $\widehat{\mathbf{f}}(z)$ cannot be more than 2ϵ larger than its minimum, $H^\star(\widehat{\mathbf{v}})$. We use this to bound the distance between $\widehat{\mathbf{f}}(z)$ and the minimizer $\widehat{\mathbf{v}}$. Observe that $\widehat{\mathbf{f}}(z)$ falls in the $[H^\star(\widehat{\mathbf{v}}) + 2\epsilon]$-level set of $H^\star$. So, we next look at a specific level set of $H^\star$.

Define

$$\tau := \min_{\mathbf{f} \in [0,1]^2 : \|\mathbf{f} - \widehat{\mathbf{v}}\|_2 = \frac{\Delta}{4}} H^\star(\mathbf{f}). \tag{D.9}$$

Observe that a minimum exists (infimum is not required) for the minimization in (D.9) because we are minimizing over the closed set $\{\mathbf{f} \in [0,1]^2 : \|\mathbf{f} - \widehat{\mathbf{v}}\|_2 = \frac{\Delta}{4}\}$ and $H^\star$ is continuous.

For any fixed $p \in (1, \infty)$, Equation (D.5) shows that $\widehat{v}_1$ is bounded away from 0. Hence, Claim D.1.4 shows that $H^\star$ is strictly convex at and in the region around $\widehat{\mathbf{v}}$. Further, $H^\star$

is convex everywhere else. Coupling this with the fact that (D.9) minimizes along points not arbitrarily close to the minimizer $\widehat{\mathbf{v}}$, we have $\tau > H^\star(\widehat{\mathbf{v}})$.

Define the level set of $H^\star$ with respect to τ:

$$\mathcal{C}_\tau = \{\mathbf{f} \in [0,1]^2 : H^\star(\mathbf{f}) \le \tau\}.$$

Claim D.1.9. *For every $\mathbf{f} \in \mathcal{C}_\tau$, we have $\|\mathbf{f} - \widehat{\mathbf{v}}\|_2 \le \frac{\Delta}{4}$.*

Define $\epsilon_o := \frac{\tau - H^\star(\widehat{\mathbf{v}})}{2}$, and set $\epsilon = \epsilon_o$. Then, set $z = \widetilde{z}_{\epsilon_o}$ as before. Applying Claim D.1.8, we obtain

$$H^\star(\widehat{\mathbf{f}}(\widetilde{z}_{\epsilon_o})) < H^\star(\widehat{\mathbf{v}}) + 2\epsilon_o = \tau.$$

In other words, $\widehat{\mathbf{f}}(\widetilde{z}_{\epsilon_o}) \in \mathcal{C}_\tau$. And applying Claim D.1.9, we obtain $\|\widehat{\mathbf{f}}(\widetilde{z}_{\epsilon_o}) - \widehat{\mathbf{v}}\|_2 \le \frac{\Delta}{4}$, completing part (i).

This implies that $\|\widehat{\mathbf{f}}(\widetilde{z}_{\epsilon_o}) - \widehat{\mathbf{v}}\|_\infty \le \frac{\Delta}{4}$, which means

$$\left|\widehat{f}_1(\widetilde{z}_{\epsilon_o}) - \widehat{v}_1\right| \le \frac{\Delta}{4} \quad \text{and} \quad \left|\widehat{f}_2(\widetilde{z}_{\epsilon_o}) - \widehat{v}_2\right| \le \frac{\Delta}{4}. \tag{D.10}$$

Using these properties, we have

$$\begin{aligned}
\widehat{f}_1(\widetilde{z}_{\epsilon_o}) &\le \widehat{v}_1 + \frac{\Delta}{4} \\
&= \widehat{v}_2 - \Delta + \frac{\Delta}{4} \\
&\le \widehat{f}_2(\widetilde{z}_{\epsilon_o}) + \frac{\Delta}{4} - \Delta + \frac{\Delta}{4} = \widehat{f}_2(\widetilde{z}_{\epsilon_o}) - \frac{\Delta}{2},
\end{aligned}$$

where the first inequality holds because of the first part of (D.10), the equality holds because $\Delta = \widehat{v}_2 - \widehat{v}_1$ and the second inequality holds because of the second part of (D.10). Therefore, for $z = \widetilde{z}_{\epsilon_o} > 1$, the aggregate scores of the two papers are such that

$$\widehat{f}_1(\widetilde{z}_{\epsilon_o}) < \widehat{f}_2(\widetilde{z}_{\epsilon_o}),$$

violating efficiency.

Proof of Claim D.1.1 Take arbitrary $\mathbf{f}, \mathbf{g} \in \mathbb{R}^2$ with $\mathbf{f} \neq \mathbf{g}$, and let $\theta \in (0,1)$. We show that $G_z(\theta\mathbf{f} + (1-\theta)\mathbf{g}) < \theta G_z(\mathbf{f}) + (1-\theta)G_z(\mathbf{g})$. For this, we will first show that either (i) $[(z,0) - \mathbf{f}]$ is not parallel to $[(z,0) - \mathbf{g}]$, (ii) $[(0,1) - \mathbf{f}]$ is not parallel to $[(0,1) - \mathbf{g}]$ or (iii) $\mathbf{f}$ is not parallel to $\mathbf{g}$. For the sake of contradiction, assume that this is not true. That is, assume $[(z,0) - \mathbf{f}]$ is parallel to $[(z,0) - \mathbf{g}]$, $[(0,1) - \mathbf{f}]$ is parallel to $[(0,1) - \mathbf{g}]$, and $\mathbf{f}$ is parallel to $\mathbf{g}$. This implies that

$$\begin{bmatrix} z - f_1 \\ -f_2 \end{bmatrix} = r \begin{bmatrix} z - g_1 \\ -g_2 \end{bmatrix}, \qquad \begin{bmatrix} -f_1 \\ 1 - f_2 \end{bmatrix} = s \begin{bmatrix} -g_1 \\ 1 - g_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = t \begin{bmatrix} g_1 \\ g_2 \end{bmatrix},$$

where $r, s, t \in \mathbb{R}$ [1]. Note that, none of r, s, t can be 1 because $\mathbf{f} \neq \mathbf{g}$. The second equation tells us that $f_1 = sg_1$ and the third one tells us that $f_1 = tg_1$. So, either $f_1 = g_1 = 0$ or

[1] A boundary case not captured here is when $\mathbf{g}$ is exactly one of the points $(z,0), (0,1)$ or $(0,0)$, leading to $1/r, 1/s$ or $1/t$ being zero respectively. But for this case, it is easy to prove that the other two pairs of vectors cannot be parallel unless $\mathbf{f} = \mathbf{g}$.

$s = t$. But from the first equation, $z - f_1 = rz - rg_1$. So if $f_1 = g_1 = 0$, it says that $r = 1$ which is not possible. Therefore, $s = t$. The third equation now tells us that $f_2 = tg_2 = sg_2$. But, the second equation gives us $1 - f_2 = s - sg_2$, which implies that $s = 1$. But again this is not possible, leading to a contradiction. Therefore, at least one of (i), (ii) and (iii) is true.

L_p norm with $p \in (1, \infty)$ is a convex norm, i.e. for any $x, y \in \mathbb{R}^2$,

$$\|\theta x + (1-\theta)y\|_p \leq \theta\|x\|_p + (1-\theta)\|y\|_p. \tag{D.11}$$

Further, since $p \in (1, \infty)$, the inequality in (D.11) is strict if x is not parallel to y. For our objective (in Equation (D.1)),

$$\begin{aligned} G_z(\theta\mathbf{f} + (1-\theta)\mathbf{g}) = &\left\|\theta[(z,0) - \mathbf{f}] + (1-\theta)[(z,0) - \mathbf{g}]\right\|_p \\ &+ \left\|\theta[(0,1) - \mathbf{f}] + (1-\theta)[(0,1) - \mathbf{g}]\right\|_p \\ &+ \left\|\theta\mathbf{f} + (1-\theta)\mathbf{g}\right\|_p. \end{aligned} \tag{D.12}$$

Because of convexity of the L_p norm, each of the three terms on the RHS of Equation (D.12) satisfies inequality (D.11). Further, because at least one of the pair of vectors in the three terms is not parallel (since either (i), (ii) or (iii) is true), at least one of them gives us a strict inequality. Therefore we obtain

$$G_z(\theta\mathbf{f} + (1-\theta)\mathbf{g}) < \theta G_z(\mathbf{f}) + (1-\theta)G_z(\mathbf{g}).$$

■

Proof of Claim D.1.2 The claim has four parts: (i) $\widehat{f}_1(z) \geq 0$, (ii) $\widehat{f}_1(z) \leq 1$, (iii) $\widehat{f}_2(z) \geq 0$ and (iv) $\widehat{f}_2(z) \leq 1$. Observe that parts (i), (iii) and (iv) are more intuitive, since they show that the aggregate score of a paper is no higher than the maximum score given to it, and no lower than the minimum score given to it. Part (ii) on the other hand is stronger; even though paper 1 has a score of $z > 1$ given to it, this part shows that $\widehat{f}_1(z) \leq 1$ (which is much tighter than an upper bound of z, especially when z is large). We prove the simpler parts (i), (iii) and (iv) first.

For the sake of contradiction, suppose $\widehat{f}_1(z) < 0$. Then

$$\begin{aligned} G_z(\widehat{f}_1(z), \widehat{f}_2(z)) &= \left[|z - \widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |1 - \widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} \\ &> \left[|z|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[0 + |1 - \widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[0 + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} = G_z(0, \widehat{f}_2(z)), \end{aligned}$$

contradicting the fact that $(\widehat{f}_1(z), \widehat{f}_2(z))$ is optimal. Therefore, $\widehat{f}_1(z) \geq 0$, completing proof of (i). Similarly, if $\widehat{f}_2(z) < 0$, we can show that $G_z(\widehat{f}_1(z), \widehat{f}_2(z)) > G_z(\widehat{f}_1(z), 0)$, violating optimality. Therefore, $\widehat{f}_2(z) \geq 0$, completing proof of (iii).

Next, for the sake of contradiction assume that $\widehat{f}_2(z) > 1$. Then

$$\begin{aligned} G_z(\widehat{f}_1(z), \widehat{f}_2(z)) &= \left[|z - \widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |1 - \widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} \\ &> \left[|z - \widehat{f}_1(z)|^p + 1\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + 0\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + 1\right]^{\frac{1}{p}} = G_z(\widehat{f}_1(z), 1), \end{aligned}$$

contradicting the fact that $(\widehat{f}_1(z), \widehat{f}_2(z))$ is optimal. Therefore, we also have $\widehat{f}_2(z) \leq 1$, completing proof of (iv).

Finally, we prove the more non-intuitive part, (ii). Suppose for the sake of contradiction, $\widehat{f}_1(z) > 1$. Then,

$$\begin{aligned} G_z(\widehat{f}_1(z), \widehat{f}_2(z)) &= \left[|z - \widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |1 - \widehat{f}_2(z)|^p\right]^{\frac{1}{p}} + \left[|\widehat{f}_1(z)|^p + |\widehat{f}_2(z)|^p\right]^{\frac{1}{p}} \\ &\geq |z - \widehat{f}_1(z)| + |\widehat{f}_1(z)| + |\widehat{f}_1(z)| \\ &\geq z + |\widehat{f}_1(z)|, \end{aligned}$$

where the first inequality comes from the fact that the L_p norm of each vector is at least as high as the absolute value of its first element, and the second inequality follows from the triangle inequality. Using the assumption that $\widehat{f}_1(z) > 1$, we obtain

$$G_z(\widehat{f}_1(z), \widehat{f}_2(z)) > z + 1 = G_z(0, 0),$$

contradicting the fact that $(\widehat{f}_1(z), \widehat{f}_2(z))$ is optimal. Therefore, $\widehat{f}_1(z) \leq 1$, completing the proof. ■

Proof of Claim D.1.3 Take any arbitrary $f_1 \in [0, 1]$ and $f_2 \in [0, 1]$. Subtracting Equations (D.3) and (D.4) we obtain

$$H_z(f_1, f_2) - H^\star(f_1, f_2) = \left[\big(z - f_1\big)^p + f_2^p\right]^{\frac{1}{p}} - \big(z - f_1\big). \tag{D.13}$$

Observe that since $f_2 \geq 0$, the RHS of Equation (D.13) is non-negative. Hence, the equation does not change on using an absolute value, i.e.,

$$|H_z(f_1, f_2) - H^\star(f_1, f_2)| = \left[\big(z - f_1\big)^p + f_2^p\right]^{\frac{1}{p}} - \big(z - f_1\big). \tag{D.14}$$

To prove the required result, we take a small detour and define $\phi(x) = (x^p + f_2^p)^{\frac{1}{p}} - x$. We show that $\phi(x) \to 0$ as $x \to \infty$. For this, rewrite $\phi(x)$ as follows

$$\phi(x) = x\left(1 + \frac{f_2^p}{x^p}\right)^{\frac{1}{p}} - x = \frac{\left(1 + \frac{f_2^p}{x^p}\right)^{\frac{1}{p}} - 1}{\frac{1}{x}}.$$

Taking the limit of x to infinity, we have

$$\lim_{x\to\infty} \phi(x) = \lim_{x\to\infty} \frac{\left(1 + \frac{f_2^p}{x^p}\right)^{\frac{1}{p}} - 1}{\frac{1}{x}}. \tag{D.15}$$

Observe that for both the numerator and denominator in the RHS of Equation (D.15), we have

$$\lim_{x\to\infty} \left\{\left(1 + \frac{f_2^p}{x^p}\right)^{\frac{1}{p}} - 1\right\} = 0 \quad \text{and} \quad \lim_{x\to\infty} \left\{\frac{1}{x}\right\} = 0.$$

Hence, applying L'Hospital's rule on equation (D.15) gives us

$$\begin{aligned}\lim_{x\to\infty}\phi(x) &= \lim_{x\to\infty}\frac{-\frac{f_2^p}{x^{p+1}}\left(1+\frac{f_2^p}{x^p}\right)^{\frac{1}{p}-1}}{-\frac{1}{x^2}}\\ &= \lim_{x\to\infty}\left\{\frac{f_2^p}{x^{p-1}}\left(1+\frac{f_2^p}{x^p}\right)^{\frac{1}{p}-1}\right\}\\ &= \left[\lim_{x\to\infty}\frac{f_2^p}{x^{p-1}}\right] * \left[\lim_{x\to\infty}\left(1+\frac{f_2^p}{x^p}\right)^{\frac{1}{p}-1}\right]\\ &= 0 * 1 = 0,\end{aligned}$$

where $\left[\lim_{x\to\infty}\frac{f_2^p}{x^{p-1}}\right] = 0$ because $p > 1$. Hence, we proved the required result, $\lim_{x\to\infty}\phi(x) = 0$. Going back to Equation (D.14), we rewrite it as

$$|H_z(f_1,f_2) - H^\star(f_1,f_2)| = \left[\left(z-f_1\right)^p + f_2^p\right]^{\frac{1}{p}} - \left(z-f_1\right) = \phi(z-f_1).$$

Taking the limit of z to infinity, we obtain

$$\lim_{z\to\infty}|H_z(f_1,f_2) - H^\star(f_1,f_2)| = \lim_{z\to\infty}\phi(z-f_1) = \lim_{t\to\infty}\phi(t) = 0, \tag{D.16}$$

where the second step follows by setting $t = z - f_1$. Equation (D.16) implies that

$$\lim_{z\to\infty} H_z(f_1,f_2) = H^\star(f_1,f_2).$$

■

Proof of Claim D.1.4 In the region $[0,1]^2$, using (D.4), the function $H^\star$ can be written as

$$H^\star(f_1,f_2) = -f_1 + \|(0,1)-(f_1,f_2)\|_p + \|(f_1,f_2)\|_p - 1. \tag{D.17}$$

Observe that each term on the RHS of (D.17) is a convex function of $\mathbf{f}$. Hence, their sum is also convex in $\mathbf{f}$.

The proof of strict convexity closely follows the proof of claim D.1.1. Take arbitrary $\mathbf{f},\mathbf{g} \in (0,1]\times[0,1]$ with $\mathbf{f}\neq\mathbf{g}$, and let $\theta\in(0,1)$. We show that $H^\star(\theta\mathbf{f}+(1-\theta)\mathbf{g}) < \theta H^\star(\mathbf{f}) + (1-\theta)H^\star(\mathbf{g})$. For this, we will first show that either (i) $[(0,1)-\mathbf{f}]$ is not parallel to $[(0,1)-\mathbf{g}]$ or (ii) $\mathbf{f}$ is not parallel to $\mathbf{g}$. For the sake of contradiction, assume that this is not true. That is, assume $[(0,1)-\mathbf{f}]$ is parallel to $[(0,1)-\mathbf{g}]$, and $\mathbf{f}$ is parallel to $\mathbf{g}$. This implies that

$$\begin{bmatrix}-f_1\\1-f_2\end{bmatrix} = r\begin{bmatrix}-g_1\\1-g_2\end{bmatrix} \quad\text{and}\quad \begin{bmatrix}f_1\\f_2\end{bmatrix} = s\begin{bmatrix}g_1\\g_2\end{bmatrix},$$

where $r,s\in\mathbb{R}$. Note that, neither r nor s can be 1 because $\mathbf{f}\neq\mathbf{g}$. The first equation tells us that $f_1 = rg_1$ and the second one tells us that $f_1 = sg_1$. And since $g_1\neq 0$, this implies

that $r = s$. The second part of the second equation now tells us that $f_2 = sg_2 = rg_2$. The second part of the first equation becomes $1 - f_2 = r - rg_2$ which implies that $r = 1$, leading to a contradiction. Therefore, at least one of (i) and (ii) is true.

Recall, L_p norm with $p \in (1, \infty)$ is a convex norm, i.e. for any $x, y \in \mathbb{R}^2$,

$$\|\theta x + (1-\theta)y\|_p \leq \theta\|x\|_p + (1-\theta)\|y\|_p. \tag{D.18}$$

And since $p \in (1, \infty)$, the inequality in (D.18) is strict if x is not parallel to y. For $H^\star$ (using Equation (D.17)),

$$\begin{aligned} H^\star(\theta\mathbf{f} + (1-\theta)\mathbf{g}) = &-\theta f_1 - (1-\theta)g_1 \\ &+ \left\|\theta[(0,1) - \mathbf{f}] + (1-\theta)[(0,1) - \mathbf{g}]\right\|_p \\ &+ \left\|\theta\mathbf{f} + (1-\theta)\mathbf{g}\right\|_p - 1. \end{aligned} \tag{D.19}$$

Because of convexity of the L_p norm, both the third and fourth term on the RHS of Equation (D.19) satisfy inequality (D.18). Further, because at least one of the pair of vectors in these two terms is not parallel (since either (i) or (ii) is true), at least one of them gives us a strict inequality. Therefore we obtain

$$H^\star(\theta\mathbf{f} + (1-\theta)\mathbf{g}) < \theta H^\star(\mathbf{f}) + (1-\theta)H^\star(\mathbf{g}).$$

■

Proof of Claim D.1.5 To compute the minimizer of $H^\star$, we compute its gradients with respect to f_1 and f_2. Using Equation (D.4), the partial derivative with respect to f_1 is

$$\frac{\partial H^\star}{\partial f_1} = -1 + f_1^{p-1}\left[f_1^p + (1-f_2)^p\right]^{\frac{1}{p}-1} + f_1^{p-1}\left[f_1^p + f_2^p\right]^{\frac{1}{p}-1} \tag{D.20}$$

and with respect to f_2 is

$$\frac{\partial H^\star}{\partial f_2} = 0 - (1-f_2)^{p-1}\left[f_1^p + (1-f_2)^p\right]^{\frac{1}{p}-1} + f_2^{p-1}\left[f_1^p + f_2^p\right]^{\frac{1}{p}-1}. \tag{D.21}$$

Observe that at $f_2 = \frac{1}{2}$, irrespective of the value of f_1, the partial derivative (D.21) is

$$\left.\frac{\partial H^\star}{\partial f_2}\right|_{f_2=\frac{1}{2}} = -\frac{1}{2^{p-1}}\left[f_1^p + \frac{1}{2^p}\right]^{\frac{1}{p}-1} + \frac{1}{2^{p-1}}\left[f_1^p + \frac{1}{2^p}\right]^{\frac{1}{p}-1} = 0.$$

So, set $\widehat{v}_2 = \frac{1}{2}$. Next, we find $\widehat{v}_1$ such that the other derivative (D.20) is also zero at $\widehat{\mathbf{v}} = (\widehat{v}_1, \widehat{v}_2)$. Setting (D.20) to zero at $\widehat{\mathbf{v}}$, we obtain

$$\begin{aligned} \left.\frac{\partial H^\star}{\partial f_1}\right|_{\mathbf{f}=\widehat{\mathbf{v}}} = 0 &= -1 + \widehat{v}_1^{p-1}\left[\widehat{v}_1^p + \frac{1}{2^p}\right]^{\frac{1}{p}-1} + \widehat{v}_1^{p-1}\left[\widehat{v}_1^p + \frac{1}{2^p}\right]^{\frac{1}{p}-1} \\ \implies 1 &= 2\widehat{v}_1^{p-1}\left[\widehat{v}_1^p + \frac{1}{2^p}\right]^{\frac{1}{p}-1} \end{aligned}$$

$$
\begin{aligned}
\implies \left[\widehat{v}_1^p + \frac{1}{2^p}\right]^{1-\frac{1}{p}} &= 2\widehat{v}_1^{p-1} \\
\implies \left[\widehat{v}_1^p + \frac{1}{2^p}\right]^{p-1} &= 2^p \widehat{v}_1^{p(p-1)} \\
\implies \widehat{v}_1^p + \frac{1}{2^p} &= 2^{\frac{p}{p-1}} \widehat{v}_1^p \\
\implies \frac{1}{2^p} &= \widehat{v}_1^p \left(2^{\frac{p}{p-1}} - 1\right) \\
\therefore \widehat{v}_1 &= \frac{1}{2}\left[\frac{1}{(2^{\frac{p}{p-1}} - 1)}\right]^{\frac{1}{p}}.
\end{aligned}
$$

Hence, $\nabla_{\mathbf{f}} H^\star(\mathbf{f}) = \mathbf{0}$ at $\widehat{\mathbf{v}}$. And since $H^\star$ is convex in $[0,1]^2$ by Claim D.1.4, $\widehat{\mathbf{v}}$ is the minimizer in this region. ■

Proof of Claim D.1.6 For any $p > 1$, we know

$$\frac{p}{p-1} > 1.$$

This implies that

$$2^{\frac{p}{p-1}} - 1 > 1 \quad \text{and hence} \quad \left[\frac{1}{2^{\frac{p}{p-1}} - 1}\right]^{\frac{1}{p}} < 1.$$

Finally, using the values from Claim D.1.5, we obtain

$$\widehat{v}_1 < \widehat{v}_2.$$

■

Proof of Claim D.1.7 Let $z \in \mathcal{Z}_\epsilon(1,1)$. Pick an arbitrary $(f_1, f_2) \in [0,1]^2$. As in the proof of Claim D.1.3, on subtracting Equations (D.3) and (D.4), and taking an absolute value, we obtain Equation (D.14), that is,

$$|H_z(f_1, f_2) - H^\star(f_1, f_2)| = \left[\big(z - f_1\big)^p + f_2^p\right]^{\frac{1}{p}} - \big(z - f_1\big). \tag{D.22}$$

Combining Equation (D.22) with the fact that $0 \le f_2 \le 1$, we obtain

$$|H_z(f_1, f_2) - H^\star(f_1, f_2)| \le \left[\big(z - f_1\big)^p + 1\right]^{\frac{1}{p}} - \big(z - f_1\big). \tag{D.23}$$

Now, define $\psi(x) = (x^p + 1)^{\frac{1}{p}} - x$. We show that $\psi(x)$ is a non-increasing function for $x \ge 0$. Computing the derivative, we have

$$\frac{d\psi(x)}{dx} = x^{p-1}\left(x^p + 1\right)^{\frac{1}{p}-1} - 1 = \left(\frac{x^p}{x^p + 1}\right)^{\frac{p-1}{p}} - 1 \le 0$$

for $x \geq 0$, showing that it is a non-increasing function. Going back to Equation (D.23), we know that $f_1 \leq 1$. Therefore, $\big(z - f_1\big) \geq \big(z - 1\big) \geq 0$. Using the fact that ψ is a non-increasing function, we obtain $\psi\big(z - f_1\big) \leq \psi\big(z - 1\big)$, which on expansion gives us

$$\Big[\big(z - f_1\big)^p + 1\Big]^{\frac{1}{p}} - \big(z - f_1\big) \leq \Big[\big(z - 1\big)^p + 1\Big]^{\frac{1}{p}} - \big(z - 1\big) = |H_z(1,1) - H^\star(1,1)|. \quad \text{(D.24)}$$

Combining Equations (D.23) and (D.24), and the fact that $z \in \mathcal{Z}_\epsilon(1,1)$, we obtain

$$|H_z(f_1, f_2) - H^\star(f_1, f_2)| \leq |H_z(1,1) - H^\star(1,1)| < \epsilon.$$

Hence, $z \in \mathcal{Z}_\epsilon(f_1, f_2)$. ■

Proof of Claim D.1.8 The proof follows using three facts:

1. Equation (D.8) for $\widehat{\mathbf{f}}(z)$ says that $H^\star(\widehat{\mathbf{f}}(z)) < H_z(\widehat{\mathbf{f}}(z)) + \epsilon$.
2. Because $\widehat{\mathbf{f}}(z)$ is the minimizer of H_z, we have $H_z(\widehat{\mathbf{f}}(z)) \leq H_z(\widehat{\mathbf{v}})$.
3. For $\widehat{\mathbf{v}}$, Equation (D.8) gives us $H_z(\widehat{\mathbf{v}}) < H^\star(\widehat{\mathbf{v}}) + \epsilon$.

Putting these equations together:

$$H^\star(\widehat{\mathbf{f}}(z)) < H_z(\widehat{\mathbf{f}}(z)) + \epsilon \leq H_z(\widehat{\mathbf{v}}) + \epsilon < H^\star(\widehat{\mathbf{v}}) + 2\epsilon.$$

■

Proof of Claim D.1.9 We prove the claim by contraposition. Pick an arbitrary $\mathbf{f} \in [0,1]^2$ such that $\|\mathbf{f} - \widehat{\mathbf{v}}\|_2 > \frac{\Delta}{4}$. This means that there exists $\mathbf{g} \in [0,1]^2$ on the line joining $\mathbf{f}$ and $\widehat{\mathbf{v}}$ such that $\|\mathbf{g} - \widehat{\mathbf{v}}\|_2 = \frac{\Delta}{4}$. We could alternatively write $\mathbf{g} = \theta\mathbf{f} + (1-\theta)\widehat{\mathbf{v}}$, where $\theta \in (0,1)$. By convexity of $H^\star$,

$$H^\star(\mathbf{g}) \leq \theta H^\star(\mathbf{f}) + (1-\theta)H^\star(\widehat{\mathbf{v}}). \quad \text{(D.25)}$$

By definition of τ in (D.9), we know $H^\star(\mathbf{g}) \geq \tau$. Also, we know $H^\star(\widehat{\mathbf{v}}) < \tau$. Using these in (D.25), we obtain

$$\tau < \theta H^\star(\mathbf{f}) + (1-\theta)\tau.$$

Therefore, we obtain $H^\star(\mathbf{f}) > \tau$. In summary, if $\|\mathbf{f} - \widehat{\mathbf{v}}\|_2 > \frac{\Delta}{4}$, then $H^\star(\mathbf{f}) > \tau$. Taking the contrapositive gives us the desired result. ■

D.1.2 Proof of Lemma 5.3.4

Consider $L(p,q)$ aggregation with arbitrary $q \in (1, \infty]$. We show that strategyproofness is violated. The construction for this is as follows. Suppose there is one paper a and two reviewers. The first reviewer gives the paper an overall recommendation of 1 and the second reviewer gives it an overall recommendation of 0. Let $\mathbf{x}_a$ be the (objective) criteria scores of this paper.

Let us first consider $q \in (1, \infty)$. For a function $f : \mathbb{X} \to \mathbb{Y}$, all we care about in this example is its value at $\mathbf{x}_a$. Hence, for simplicity, let f_a denote the value of function f at $\mathbf{x}_a$, i.e, $f_a := f(\mathbf{x}_a)$. Then our aggregation becomes

$$\widehat{f}_a = \operatorname*{argmin}_{f_a \in \mathbb{R}} \Big\{ |1 - f_a|^q + |f_a|^q \Big\}.$$

We claim that $f_a = 0.5$ is the unique minimizer. Observe that if $f_a = 0.5$, then the value of our objective is $0.5^q + 0.5^q < 1$ when $q \in (1, \infty)$. On the other hand, if $f_a \geq 1$ or if $f_a \leq 0$ then the value of our objective is at least 1. Hence $f_a \in (0, 1)$. By symmetry, we can restrict attention to the range $[0.5, 1)$ since if there is a minimizer in $(0, 0.5)$ then there must also be a minimizer in $(0.5, 1)$. Consequently, we rewrite the optimization problem as

$$\widehat{f}_a = \operatorname*{argmin}_{f_a \in [0.5, 1)} \Big\{ (1 - f_a)^q + f_a^q \Big\}. \tag{D.26}$$

Consider the function $h : [0.5, 1] \to \mathbb{R}$ defined by $h(x) = x^q$. This function is strictly convex (the second derivative is strictly positive in the domain) whenever $q \in (1, \infty)$. Hence from the definition of strict convexity, we have

$$0.5\big((1 - f_a)^q + f_a^q\big) > \big(0.5(1 - f_a + f_a)\big)^q = 0.5^q$$

whenever $f_a \in (0.5, 1)$. Consequently, the objective value of (D.26) is greater at $f_a \in (0.5, 1)$ than at $f_a = 0.5$. We conclude that $\widehat{f}_a = 0.5$ whenever $q \in (1, \infty)$.

When $q = \infty$, we equivalently write the optimization problem as

$$\widehat{f}_a = \operatorname*{argmin}_{f_a \in \mathbb{R}} \ \max\big(|1 - f_a|, |f_a|\big).$$

This objective has a value of 0.5 if $f_a = 0.5$ and strictly greater if $f_a \neq 0.5$. Hence, $\widehat{f}_a = 0.5$ for $q = \infty$ as well.

The true overall recommendation of reviewer 1 differs from the aggregate $\widehat{f}_a$ by 0.5 (in every L_ℓ norm). However, if reviewer 1 reported an overall recommendation of 2, then an argument identical to that above shows that the minimizer is $\widehat{g}_a = 1$. Reviewer 1 has thus successfully brought down the difference between her own true overall recommendation and the aggregate $\widehat{g}_a$ to 0. We conclude that strategyproofness is violated whenever $q \in (1, \infty]$. ■

D.1.3 Proof of Lemma 5.3.5

The construction showing that $L(\infty, 1)$ aggregation violates consensus is as follows. Suppose there are two papers, two reviewers and both reviewers review both papers. Assume that the papers have objective criteria scores $\mathbf{x}_1$ and $\mathbf{x}_2$, and that neither of these scores is pointwise greater than or equal to the other. Let the overall recommendations of the reviewers for the papers be given by the matrix

$$\mathbf{y} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix},$$

where y_{ia} denotes the overall recommendation of reviewer i for paper a. Since both reviewers give the same overall recommendation of 1 to paper 2, any aggregation method that satisfies consensus must also give paper 2 an aggregate score of 1. We show that this is not the case under $L(\infty, 1)$ aggregation.

Let f_i denote the value of function f on paper i, i.e. $f_i := f(\mathbf{x}_i)$. And let $\widehat{f_i}$ denote the aggregate score of paper i. Since we are minimizing $L(\infty, 1)$ loss, the aggregate function satisfies:

$$(\widehat{f_1}, \widehat{f_2}) \in \operatorname*{argmin}_{(f_1,f_2)\in\mathbb{R}^2} \left\{ \left\|(0,1) - (f_1, f_2)\right\|_\infty + \left\|(2,1) - (f_1, f_2)\right\|_\infty \right\}. \tag{D.27}$$

We do not have any monotonicity constraints in (D.27) because the two papers have incomparable criteria scores. Denote the objective function of (D.27) by $G(f_1, f_2)$. We can simplify this objective to

$$G(f_1, f_2) = \max(|f_1|, |f_2 - 1|) + \max(|2 - f_1|, |f_2 - 1|). \tag{D.28}$$

We claim that $(0.5, 0.5)$ is a minimizer of G. The objective function value at this point is

$$G(0.5, 0.5) = \max(0.5, 0.5) + \max(1.5, 0.5) = 0.5 + 1.5 = 2.$$

For arbitrary $(f_1, f_2) \in \mathbb{R}^2$, we have

$$\begin{aligned} G(f_1, f_2) &= \max(|f_1|, |f_2 - 1|) + \max(|2 - f_1|, |f_2 - 1|) \\ &\geq |f_1| + |2 - f_1| \\ &\geq 2 = G(0.5, 0.5), \end{aligned}$$

where the first inequality holds because the maximum of two elements is always larger than the first, and the second inequality holds by the triangle inequality. Therefore, $(0.5, 0.5)$ is a minimizer of G. The L_2 norm of this minimizer is $0.5\sqrt{2} < 1$. On the other hand, any minimizer $(\widehat{f_1}, \widehat{f_2})$ with $\widehat{f_2} = 1$ would have an L_2 norm of at least 1. It follows that such a minimizer will not be selected. In other words, $L(\infty, 1)$ aggregation would select a minimizer for which the aggregate score of paper 2 is not 1, violating consensus.[2] ■

Complete picture of minimizers For completeness, we look at the set of all minimizers of G. This is given by

$$\widehat{F} = \left\{(f_1, f_2) \mid f_1 \in [0, 2], f_2 \in [1 - \min(f_1, 2 - f_1), 1 + \min(f_1, 2 - f_1)]\right\}.$$

Pictorially, this set is given by the shaded square in Figure D.1. It is the square with vertices at $(0, 1)$, $(1, 0)$, $(2, 1)$ and $(1, 2)$.

[2]Observe that even if we used any L_k norm with $k \in (1, \infty)$ for tie-breaking, the L_k norm of $(0.5, 0.5)$ would be $0.5\sqrt[k]{2} < 1$, while the L_k norm of any minimizer $(\widehat{f_1}, 1)$ would still be at least 1, violating consensus.

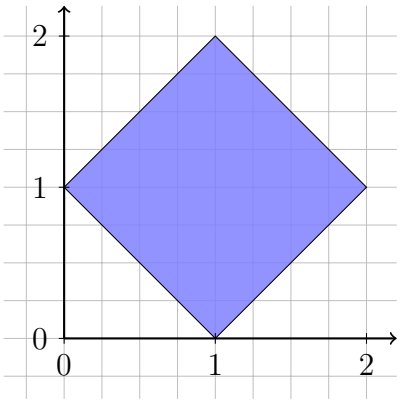


Figure D.1: The shaded region depicts the set of all minimizers of (D.27). f_1 is on the x-axis and f_2 is on the y-axis.

This shows that almost all minimizers violate consensus. For the specific tie-breaking considered, the minimizer chosen is the one with minimum L_2 norm, i.e., the projection of $(0, 0)$ onto this square. This gives us $(0.5, 0.5)$, violating consensus.

Observe that tie-breaking using minimum L_k norm, for $k \in (1, \infty]$, also chooses $(0.5, 0.5)$ as the aggregate function, violating consensus. For $k = 1$, all points on the line segment $f_1 + f_2 = 1$ $(0 \leq f_1 \leq 1)$ would be tied winners, almost all of which violate consensus. Further, even if one uses other reasonable tie-breaking schemes like maximum L_k norm, they suffer from the same issue, i.e., there is a tied winner which violates consensus.

D.2 Additional Empirical Results

We present some more empirical results in addition to those provided in the main text.

D.2.1 Influence of Varying the Hyperparameters

Although our theoretical results identify $L(1, 1)$ aggregation as the most desirable, we would like to paint a broader picture by determining how much impact the choice of p and q actually has on selected papers. To this end, we compute the overlap between the papers selected by $L(p, q)$ aggregation, for $p, q \in \{1, 2, 3\}$ (although in general p and q need not be integral, they can be real as well as ∞). Table D.1 shows the overlap between papers selected by $L(p_1, q_1)$ and $L(p_2, q_2)$, where the rows represent (p_1, q_1) and columns represent (p_2, q_2). Note that the table is symmetric. The results suggest that q has a more significant impact than p on $L(p, q)$ aggregation. For instance, $L(1, 1)$ behaves more similarly to $L(2, 1)$ and $L(3, 1)$ than to $L(1, 2)$ and $L(1, 3)$.

	1,1	1,2	1,3	2,1	2,2	2,3	3,1	3,2	3,3
1,1	100.0	87.5	82.7	96.1	88.0	82.6	92.3	87.5	82.1
1,2	87.5	100.0	94.5	88.3	94.9	93.1	87.7	94.6	92.3
1,3	82.7	94.5	100.0	84.0	92.1	95.2	83.5	91.8	94.0
2,1	96.1	88.3	84.0	100.0	89.8	84.4	95.7	89.5	84.0
2,2	88.0	94.9	92.1	89.8	100.0	94.1	89.8	98.8	93.7
2,3	82.6	93.1	95.2	84.4	94.1	100.0	84.4	94.1	98.6
3,1	92.3	87.7	83.5	95.7	89.8	84.4	100.0	89.7	84.0
3,2	87.5	94.6	91.8	89.5	98.8	94.1	89.7	100.0	93.8
3,3	82.1	92.3	94.0	84.0	93.7	98.6	84.0	93.8	100.0

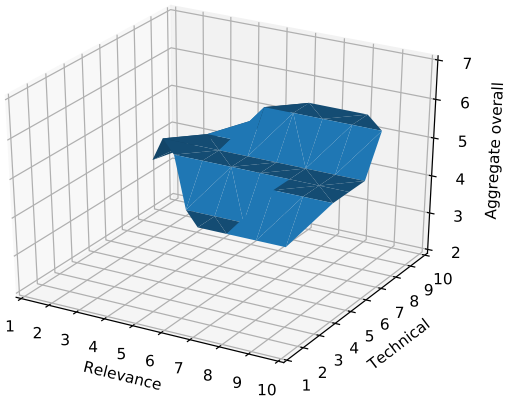
Table D.1: Percentage of overlap (in selected papers) between different $L(p, q)$ aggregation methods

D.2.2 Visualizing the Community Aggregate Mapping

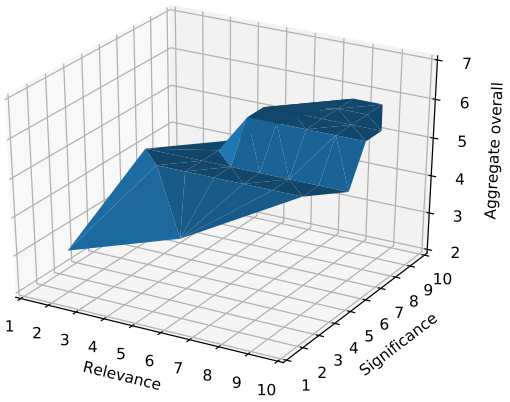
Our framework is not only useful for computing an aggregate mapping to help in acceptance decisions, but also for understanding the preferences of the community for use in subsequent modeling and research. We illustrate this application by providing some visualizations and interpretations of the aggregate function $\widetilde{f}$ obtained from $L(1, 1)$ aggregation on the IJCAI review data.

The function $\widetilde{f}$ lives in a 5-dimensional space, making it hard to visualize the entire aggregate function. Instead, we fix the values of 3 criteria at a time and plot the function in terms of the remaining two criteria. In all of the visualization and interpretation below, the fixed criteria are set to their respective (marginal) modes: For 'quality of writing' the mode is 7 (715 reviews), for 'originality' it is 6 (826 reviews), for 'relevance' it is 8 (888 reviews), for 'significance' it is 5 (800 reviews), and for 'technical quality' it is 6 (702 reviews). These plots are given in Figures D.2 and D.3.

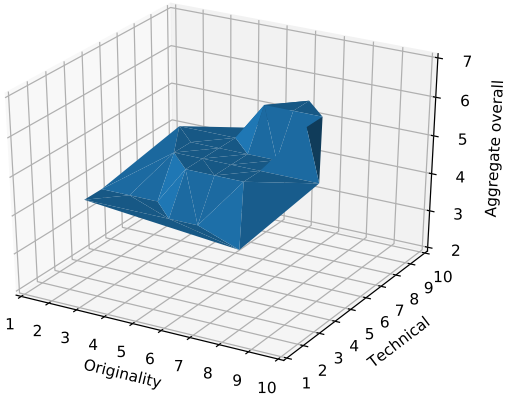
The key takeaways from this experiment are as follows. First, writing and relevance do not have a significant influence (Figure D.2e). Really bad writing or relevance is a significant downside, excellent writing or relevance is appreciated, but everything else in between in irrelevant. Second, technical quality and significance exert a high influence (Figure D.2f). Moreover, the influence is approximately linear. Third, linear models (i.e., models that are linear in the criteria) are quite popular in machine learning, and our empirical observations reveal that linear models are partially applicable to conference review data — for some criteria one may indeed assume a linear model, but not for all.



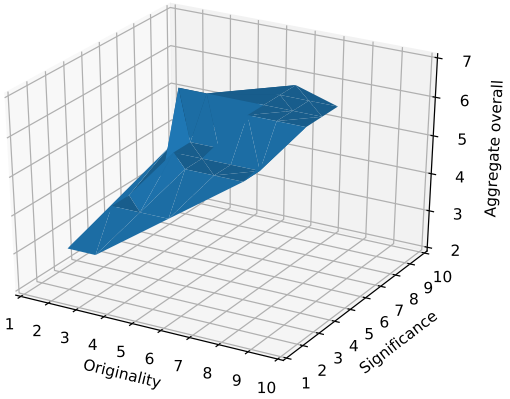
(a) Varying ‘relevance’ and ‘technical quality’



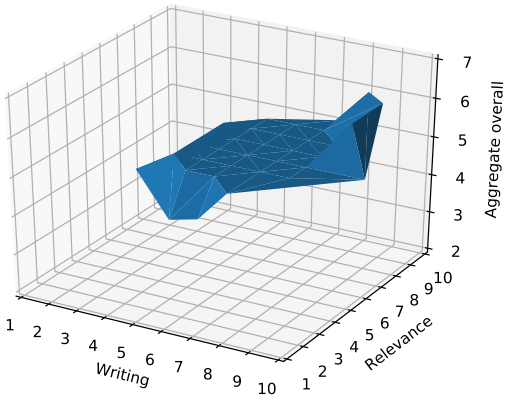
(b) Varying ‘relevance’ and ‘significance’



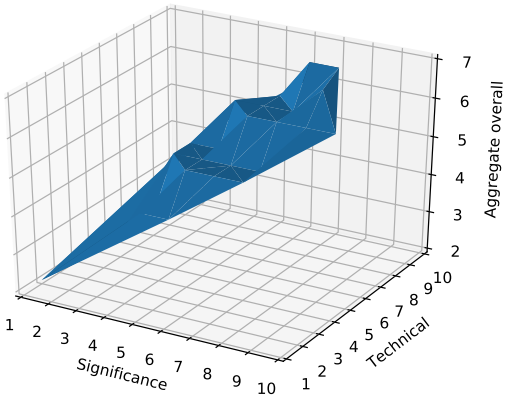
(c) Varying ‘originality’ and ‘technical quality’



(d) Varying ‘originality’ and ‘significance’

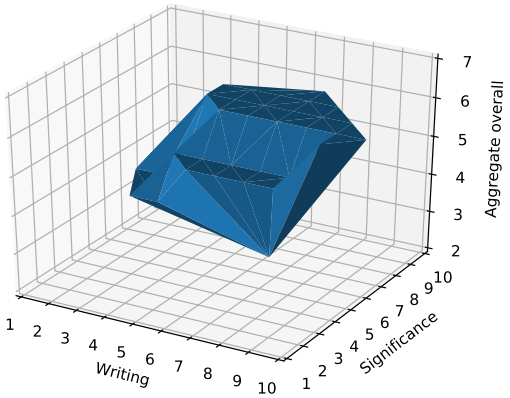


(e) Varying ‘quality of writing’ and ‘relevance’

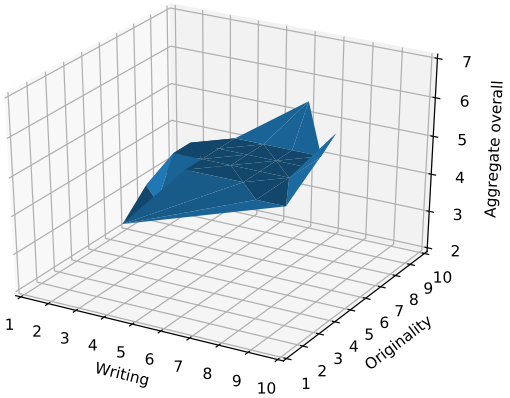


(f) Varying ‘significance’ and ‘technical quality’

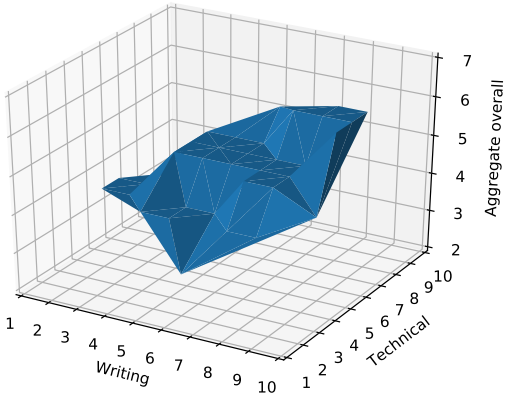
Figure D.2: Impact of varying different criteria under $L(1,1)$ aggregation



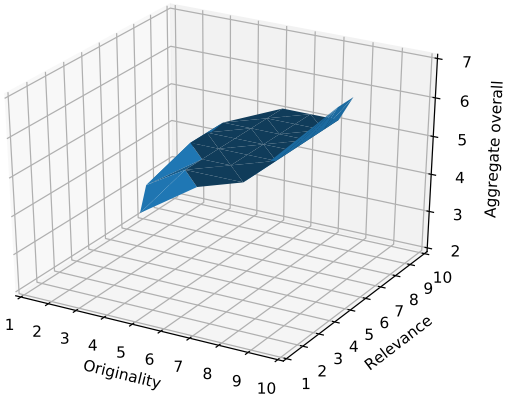
(a) Varying ‘quality of writing’ and ‘significance’



(b) Varying ‘quality of writing’ and ‘originality’



(c) Varying ‘quality of writing’ and ‘technical quality’



(d) Varying ‘originality’ and ‘relevance’

Figure D.3: Impact of varying different criteria under $L(1,1)$ aggregation (continued)

Appendix E

Omitted Proofs for Chapter 6

E.1 Appendix for Section 6.2.1

E.1.1 Proof of Lemma 6.2.1

In this proof, we also show that under the given conditions, there exists an MLE $\hat{\beta}$ satisfying

$$\|\hat{\beta}\|_\infty \leq -(|\mathcal{X}|-1)\ F^{-1}\left(2^{-K/\eta}\right),$$

where $K = \sum_{(x,y)\in\mathcal{X}^2} \#\{x \succ y\}$ and $\eta = \min_{(x,y):\#\{x\succ y\}>0} \#\{x \succ y\}$.[1]

Suppose the comparison graph $\mathcal{G}_\#$ is such that each of its connected components is strongly connected. First, we show that moving any connected component (keeping all other distances fixed) does not change the likelihood. In particular, let C be an arbitrary connected component that does not have the reference alternative r. The likelihood function can then be rewritten as

$$\mathcal{L}(\beta) = \sum_{x,y\in C} \#\{x \succ y\} \log F(\beta_x - \beta_y) + \sum_{x,y\notin C} \#\{x \succ y\} \log F(\beta_x - \beta_y),$$

as there are no edges between C and its complement. For any vector $\beta \in \mathcal{D}$, define $\beta^\Delta \in \mathcal{D}$ for any $\Delta \in \mathbb{R}$ as follows

$$\beta_x^\Delta = \begin{cases} \beta_x + \Delta & ;\ \text{if } x \in C \\ \beta_x & ;\ \text{otherwise.} \end{cases}$$

That is, β^Δ is the same as β, except with utilities changed by the constant Δ for C. The likelihood at this point for any Δ is

$$\mathcal{L}(\beta^\Delta) = \sum_{x,y\in C} \#\{x \succ y\} \log F\left(\beta_x + \Delta - \beta_y - \Delta\right) + \sum_{x,y\notin C} \#\{x \succ y\} \log F(\beta_x - \beta_y) = \mathcal{L}(\beta).$$

[1] That is, K denotes the total number of comparisons in the dataset, and η denotes the smallest positive comparison number in it (or equivalently, the smallest positive weight in $\mathcal{G}_\#$). Also note that, F^{-1} exists in $(0,1)$ as F is strictly monotonic and continous.

Hence, adding any Δ to a connected component does not affect the likelihood. In particular, for any maximizer, we could set Δ such that an alternative (of choice) in the connected component C has zero beta value, giving us a new maximizer. And this holds for every connected component. Hence, we just need to consider β vectors which have a reference alternative in each of the connected components in order to find a maximizer. Let $r_1, r_2, \ldots, r_k$ denote the references we set in each of the connected components $C_1, C_2, \ldots, C_k$ respectively (where k denotes the total number of connected components).

Define

$$B := -(|\mathcal{X}| - 1)F^{-1}\left(2^{-K/\eta}\right),$$

where K and η are as defined at the beginning of the proof. Consider an arbitrary beta vector (obeying the reference alternative constraints) with $\|\beta\|_\infty > B$. Then, there exists alternative $a \notin \{r_1, r_2, \ldots, r_k\}$ such that $|\beta_a| > B$. Without loss of generality, let $\beta_a > B$. Let C_t be the connected component that a lies in, with the reference alternative r_t. Consider all alternatives in C_t whose β value lies between that of r_t and a. The total number of these alternatives (including the end points r_t and a) is at most $|\mathcal{X}|$. Hence, the number of pairwise segments encountered starting from r_t and ending at a is at most $(|\mathcal{X}| - 1)$.[2] And since all these pairwise distances make up the total distance $\beta_a - \beta_{r_t} > B$, it implies that there exists at least one pairwise distance that is strictly larger than $B/(|\mathcal{X}| - 1)$. Let (b, c) denote the ends of this pairwise segment. That is, $b, c \in C_t$ such that $\beta_c - \beta_b > \frac{B}{|\mathcal{X}|-1}$ and there is no alternative in C_t with a β value lying in the segment (β_b, β_c). Let $\mathcal{U}$ denote the set of alternatives of C_t that lie to the left of b, i.e., $\mathcal{U} = \{x \in C_t | \beta_x \leq \beta_b\}$, and $\mathcal{V}$ be the set of alternatives of C_t that lie to the right of c, i.e., $\mathcal{V} = \{x \in C_t | \beta_x \geq \beta_c\}$. Since no alternative in C_t lies in between b and c, $(\mathcal{U}, \mathcal{V})$ is a partition of C_t. Next, as every connected component is strongly connected, C_t is also strongly connected. Hence, there has to be at least one edge going from $\mathcal{U}$ to $\mathcal{V}$ (otherwise, there would be no paths from alternatives in $\mathcal{U}$ to alternatives in $\mathcal{V}$ breaking strongly connectedness). Let this edge be given by $(u, v) \in \mathcal{U} \times \mathcal{V}$. This implies that $\beta_u \leq \beta_b$, $\beta_v \geq \beta_c$ and $\#\{u \succ v\} > 0$. Hence, we have

$$\beta_v - \beta_u \geq \beta_c - \beta_b > \frac{B}{|\mathcal{X}| - 1}.$$

The log-likelihood can be rewritten as

$$\begin{aligned}
\mathcal{L}(\beta) &= \#\{u \succ v\} \log F(\beta_u - \beta_v) + \sum_{(x,y) \neq (u,v)} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\
&\leq \#\{u \succ v\} \log F(\beta_u - \beta_v) \\
&< \#\{u \succ v\} \log F\left(-\frac{B}{|\mathcal{X}| - 1}\right), \qquad \text{(E.1)}
\end{aligned}$$

where the first inequality holds because $\#\{x \succ y\} \geq 0$ and $\log F(\beta_x - \beta_y) \leq 0$ (as $F(\cdot) \leq 1$), and the second inequality holds because $\#\{u \succ v\} > 0$, $\beta_u - \beta_v < -\frac{B}{|\mathcal{X}|-1}$ and $\log F$ is

[2] assuming all alternatives of C_t are placed on the real line according to their β values.

strictly increasing. Next, consider the log-likelihood of the zero vector. We have,

$$\mathcal{L}(0) = \sum_{x \neq y} \#\{x \succ y\} \log F(0) = K \log F(0),$$

as K is the total number of comparisons in the dataset. Recall the definition of B, we have,

$$B = -(|\mathcal{X}| - 1)F^{-1}\left(2^{-K/\eta}\right) \implies \log F\left(-\frac{B}{|\mathcal{X}| - 1}\right) = \frac{K}{\eta} \log\left(\frac{1}{2}\right).$$

Combining this with Equation (E.1), we have

$$\begin{aligned}
\mathcal{L}(\beta) &< \#\{u \succ v\} \log F\left(-\frac{B}{|\mathcal{X}| - 1}\right) \\
&= \#\{u \succ v\} \frac{K}{\eta} \log\left(\frac{1}{2}\right) \\
&\leq K \log\left(\frac{1}{2}\right) \\
&= K \log F(0) = \mathcal{L}(0),
\end{aligned}$$

where the inequality holds because $\eta = \min_{(x,y):\#\{x \succ y\} > 0} \#\{x \succ y\} \leq \#\{u \succ v\}$ and $\log(1/2) < 0$, and the next equality holds as $F(0) = 1/2$. Hence, this shows that $\mathcal{L}(\beta) < \mathcal{L}(0)$ for any β with $\|\beta\|_\infty > B$. In other words, such a β vector cannot be a maximizer of $\mathcal{L}$. Therefore, to maximize $\mathcal{L}(\beta)$ we just need to consider β vectors in $[-B, B]^{|\mathcal{X}|-k}$. And since this is a closed space, a maximizer always exists. Further, this maximizer satisfies $\|\hat{\beta}\|_\infty \leq B$.

Next, to prove the converse of the theorem statement, suppose there exists a connected component of $\mathcal{G}_\#$ that is not strongly connected. Denote this connected component by C. Consider all the strongly connected components of C; they form a DAG (as the condensation of a graph is always acyclic). Hence, there exists a strongly connected component in this DAG that has no incoming edge (from the rest of C). Let this strongly connected component be denoted by S. Further, as C itself is a connected component, this implies that there exists at least one edge going from S to $C \setminus S$. Putting all this together, we have strongly connected component S such that there is no (incoming) edge from $\mathcal{X} \setminus S$ to S, and there is at least one (outgoing) edge from S to $\mathcal{X} \setminus S$. Now, suppose for the sake of contradiction that $\mathcal{L}(\beta)$ has a maximizer. And, let $\hat{\beta}$ denote an MLE. The log-likelihood can be written as

$$\begin{aligned}
\mathcal{L}(\beta) = &\sum_{x,y \in S} \#\{x \succ y\} \log F(\beta_x - \beta_y) + \sum_{x \in S, y \notin S} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\
&+ \sum_{x,y \notin S} \#\{x \succ y\} \log F(\beta_x - \beta_y).
\end{aligned}$$

Consider another beta vector $\tilde{\beta} \in \mathcal{D}$ that is the same as $\hat{\beta}$ except that it has beta values

increased by a constant for alternatives in S.[3] For instance,

$$\tilde{\beta}_x = \begin{cases} \hat{\beta}_x + 1 & ; \text{if } x \in S \\ \hat{\beta}_x & ; \text{otherwise.} \end{cases}$$

The likelihood at this point is

$$\begin{aligned}
\mathcal{L}(\tilde{\beta}) &= \sum_{x,y \in S} \#\{x \succ y\} \log F(\hat{\beta}_x + 1 - \hat{\beta}_y - 1) + \sum_{x \in S, y \notin S} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y + 1) \\
&\quad + \sum_{x,y \notin S} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) \\
&> \sum_{x,y \in S} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) + \sum_{x \in S, y \notin S} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) \\
&\quad + \sum_{x,y \notin S} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) \\
&= \mathcal{L}(\hat{\beta}),
\end{aligned}$$

where the inequality holds because $\#\{x \succ y\} \geq 0$, $\log F(\hat{\beta}_x - \hat{\beta}_y + 1) > \log F(\hat{\beta}_x - \hat{\beta}_y)$ for all $(x, y) \in S \times S^C$ as $\log F$ is strictly increasing, and there exists at least one $(x, y) \in S \times S^C$ with $\#\{x \succ y\} > 0$ (because of the presence of the outgoing edge from S to S^C). This leads to a contradiction as $\tilde{\beta}$ has strictly higher likelihood than the MLE $\hat{\beta}$. Therefore, an MLE does not exist. □

E.1.2 Proof of Lemma 6.2.2

Let a be an alternative such that there is exactly one other alternative b for which $\#\{a \succ b\} + \#\{b \succ a\} > 0$. The log-likelihood function is

$$\begin{aligned}
\mathcal{L}(\beta) &= \sum_{(x,y)} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\
&= \left[\sum_{\substack{(x,y) \\ x \neq a, y \neq a}} \#\{x \succ y\} \log F(\beta_x - \beta_y) \right] + \#\{a \succ b\} \log F(\beta_a - \beta_b) + \#\{b \succ a\} \log F(\beta_y - \beta_x) \\
&= \mathcal{G}(\beta_{-a}) + \#\{a \succ b\} \log F(\beta_a - \beta_b) + \#\{b \succ a\} \log F(\beta_b - \beta_a),
\end{aligned}$$

where $\mathcal{G}$ is the part of the likelihood function not containing β_a. Maximizing $\mathcal{L}(\beta)$ is equivalent to first maximizing with respect to β_a and then with respect to the rest, β_{-a}.[4]

[3] In the case when S has the reference alternative r, the exact effect can be achieved by instead decreasing the beta values of all alternatives in $\mathcal{X} \setminus S$ by the same constant.

[4] In the case when a was set as the reference, we could always perform this optimization by placing the reference on some other alternative, and then shifting the complete learned vector back such that a is the reference again. Observe that this does not affect the learned distance of $(\hat{\beta}_a - \hat{\beta}_b)$, for which we are proving the desired property.

Hence, we maximize

$$\#\{a \succ b\} \log F(\beta_a - \beta_b) + \#\{b \succ a\} \log F(\beta_b - \beta_a) \tag{E.2}$$

with respect to β_a.

Claim E.1.1 (Coin flip likelihood)**.** *For $h, t > 0$ and $p \in (0,1)$, the function $f(p) = h \cdot log(p) + t \cdot log(1-p)$ is strictly concave with the maximum uniquely attained at*

$$\hat{p} = \frac{h}{h+t}$$

Proof. $f'(p) = \frac{h}{p} - \frac{t}{1-p}$, and $f''(p) = -\frac{h}{p^2} - \frac{t}{(1-p)^2}$. Hence, $f''(p) < 0$ for all $p \in (0,1)$ making f a strictly concave function. Further, $f'(\hat{p}) = 0$. Hence, $\hat{p}$ as defined in the claim is the point where the maximum is attained. □

Equation (E.2) can be rewritten as

$$\#\{a \succ b\} \log F(\beta_a - \beta_b) + \#\{b \succ a\} \log F(\beta_b - \beta_a) = f(F(\beta_a - \beta_b)),$$

where f is the function from Claim E.1.1 with $h = \#\{a \succ b\} > 0$ and $t = \#\{b \succ a\} > 0$, as $F(\beta_b - \beta_a) = 1 - F(\beta_a - \beta_b)$. Applying Claim E.1.1, we have

$$f(F(\beta_a - \beta_b)) \leq f(\hat{p}),$$

for all β_a, β_b, where $\hat{p} = \frac{\#\{a \succ b\}}{\#\{a \succ b\} + \#\{b \succ a\}}$. Further, this upper bound can be achieved by setting $F(\beta_a - \beta_b) = \hat{p}$, which is possible as F is invertible in $(0,1)$ by strict monotonicty and continuity. Therefore, Equation (E.2) is uniquely maximized at $\beta_a = \beta_b + F^{-1}(\hat{p})$. And hence, every MLE satisfies

$$\hat{\beta}_a = \hat{\beta}_b + F^{-1}\left(\frac{\#\{a \succ b\}}{\#\{a \succ b\} + \#\{b \succ a\}}\right) = \hat{\beta}_b + \delta(a,b).$$

□

E.1.3 Proof of Lemma 6.2.3

The initial part of this proof is similar to the proof of Lemma 6.2.1. Let B denote the bound $|\mathcal{X}| \cdot \max_{(x,y)} \delta(x,y)$. And, recall that r denotes the alternative set as the reference, i.e. $\beta_r = 0$. Suppose for the sake of contradiction that there exists an MLE $\hat{\beta}$ with $\|\hat{\beta}\|_\infty > B$. This implies that there exists an alternative a such that $|\hat{\beta}_a| > B$. WLOG, suppose $\hat{\beta}_a > B$. The number of alternatives whose β value lies between that of a and the reference r (including both these points) is at most $|\mathcal{X}|$. Hence, the number of pairwise segments encountered starting from r and ending at a is at most $(|\mathcal{X}| - 1)$.[5] And since all these pairwise distances make up the total distance $\hat{\beta}_a - \hat{\beta}_r > B$, it implies that there exists at least one pairwise distance that is strictly larger than $B/(|\mathcal{X}| - 1)$. Let (b,c) denote the

[5]assuming all the alternatives are placed on the real line according to their β values.

ends of this pairwise segment. That is, $\hat{\beta}_c - \hat{\beta}_b > \frac{B}{|\mathcal{X}|-1}$, and there is no alternative with a β value lying in the segment $(\hat{\beta}_b, \hat{\beta}_c)$. Construct a new beta vector $\tilde{\beta} \in \mathcal{D}$, such that $\tilde{\beta}$ is the same as $\hat{\beta}$ for alternatives to the left of alternative b, while is decreased by a small positive constant ϵ for all the other alternatives. That is,

$$\tilde{\beta}_x = \begin{cases} \hat{\beta}_x & ; \text{ if } \hat{\beta}_x \leq \hat{\beta}_b \\ \hat{\beta}_x - \epsilon & ; \text{ if } \hat{\beta}_x \geq \hat{\beta}_c. \end{cases}$$

In particular, choose ϵ such that the distance between b and c is still bigger than $\max_{(x,y)} \delta(x,y)$. This is possible because the original distance between b and c (i.e. $\hat{\beta}_c - \hat{\beta}_b$) is strictly larger than $\frac{B}{|\mathcal{X}|-1} = \frac{|\mathcal{X}|}{|\mathcal{X}|-1} \max_{(x,y)} \delta(x,y)$. Hence, one can choose $\epsilon > 0$ such that the new distance between b and c (i.e. $\tilde{\beta}_c - \tilde{\beta}_b$) is say the mid point of $\frac{|\mathcal{X}|}{|\mathcal{X}|-1} \max_{(x,y)} \delta(x,y)$ and $\max_{(x,y)} \delta(x,y)$. This would imply that we have

$$\tilde{\beta}_c - \tilde{\beta}_b > \max_{(x,y)} \delta(x,y). \tag{E.3}$$

Next, we show that in fact, $\mathcal{L}(\tilde{\beta}) > \mathcal{L}(\hat{\beta})$. The log-likelihood function is given as

$$\begin{aligned} \mathcal{L}(\beta) &= \sum_{(x,y)\in\mathcal{X}^2} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\ &= \sum_{\{x,y\}\subseteq\mathcal{X}} \Big[\#\{x \succ y\} \log F(\beta_x - \beta_y) + \#\{y \succ x\} \log F(\beta_y - \beta_x)\Big] \\ &= \sum_{\{x,y\}\subseteq\mathcal{X}} f_{xy}(F(\beta_x - \beta_y)), \end{aligned}$$

where f_{xy} is the function from Claim E.1.1 with $h = \#\{x \succ y\} > 0$ and $t = \#\{y \succ x\} > 0$. Hence, from the claim, this function f_{xy} is strictly concave with a maximum attained at $\hat{p}_{xy} = \frac{\#\{x\succ y\}}{\#\{x\succ y\}+\#\{y\succ x\}}$. Let's call $\mathcal{U}$ as the set of alternatives x with $\hat{\beta}_x \leq \hat{\beta}_b$ (i.e. the alternatives with β value unchanged), and $\mathcal{V}$ as the set of alternatives x with $\hat{\beta}_x \geq \hat{\beta}_c$ (i.e. the alternatives whose β value is decreased by ϵ). Observe that neither of these sets in empty, and they partition $\mathcal{X}$. Therefore, the log-likelihood at $\tilde{\beta}$ is

$$\begin{aligned} \mathcal{L}(\tilde{\beta}) &= \sum_{\{x,y\}\subseteq\mathcal{X}} f_{xy}\left(F(\tilde{\beta}_x - \tilde{\beta}_y)\right) \\ &= \sum_{\{x,y\}\subseteq\mathcal{U}} f_{xy}\left(F(\tilde{\beta}_x - \tilde{\beta}_y)\right) + \sum_{\{x,y\}\subseteq\mathcal{V}} f_{xy}\left(F(\tilde{\beta}_x - \tilde{\beta}_y)\right) + \sum_{(v,u)\in\mathcal{V}\times\mathcal{U}} f_{vu}\left(F(\tilde{\beta}_v - \tilde{\beta}_u)\right). \end{aligned}$$

Note that, for $x, y \in \mathcal{U}$, the distance $(\tilde{\beta}_x - \tilde{\beta}_y)$ is the same as $(\hat{\beta}_x - \hat{\beta}_y)$ as the β values are unchanged. In the case of $x, y \in \mathcal{V}$, again the distance $(\tilde{\beta}_x - \tilde{\beta}_y)$ is the same as $(\hat{\beta}_x - \hat{\beta}_y)$ as both β values (of x and y) are decreased by the same ϵ. Finally, for any pair $(v,u) \in \mathcal{V}\times\mathcal{U}$, we have $\tilde{\beta}_v - \tilde{\beta}_u = \hat{\beta}_v - \hat{\beta}_u - \epsilon$, i.e. this pairwise distance decreases by ϵ. Hence, the likelihood at $\tilde{\beta}$ becomes

$$\mathcal{L}(\tilde{\beta}) = \sum_{\{x,y\}\subseteq\mathcal{U}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{\{x,y\}\subseteq\mathcal{V}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{(v,u)\in\mathcal{V}\times\mathcal{U}} f_{vu}\left(F(\hat{\beta}_v - \hat{\beta}_u - \epsilon)\right).$$

Let's look at the terms $f_{vu}\left(F(\hat{\beta}_v - \hat{\beta}_u - \epsilon)\right)$ for $(v,u) \in \mathcal{V} \times \mathcal{U}$. We have

$$\hat{\beta}_v - \hat{\beta}_u > \hat{\beta}_v - \hat{\beta}_u - \epsilon = \tilde{\beta}_v - \tilde{\beta}_u \geq \tilde{\beta}_c - \tilde{\beta}_b > \max_{(x,y)} \delta(x,y) \geq \delta(v,u),$$

where the second inequality holds because $v \in \mathcal{V}$ is to the right of c while $u \in \mathcal{U}$ is to the left of b, and the third inequality holds from Equation (E.3). Rewriting this equation keeping only the main components, we have

$$\hat{\beta}_v - \hat{\beta}_u > \tilde{\beta}_v - \tilde{\beta}_u > \delta(v,u).$$

As F is a strictly increasing function, applying it to this equation gives us

$$F(\hat{\beta}_v - \hat{\beta}_u) > F(\tilde{\beta}_v - \tilde{\beta}_u) > F(\delta(v,u)) = \hat{p}_{vu},$$

where the equality holds by definition of the perfect-fit distance and $\hat{p}_{vu}$. Hence, by changing from $F(\hat{\beta}_v - \hat{\beta}_u)$ to $F(\tilde{\beta}_v - \tilde{\beta}_u)$, we move closer to the maxima of f_{vu} (or alternatively, $F(\tilde{\beta}_v - \tilde{\beta}_u)$ is a convex combination of $F(\hat{\beta}_v - \hat{\beta}_u)$ and the maxima $\hat{p}_{vu}$). But, as f_{vu} is strictly concave, it means that this change leads to an increase in its value. That is,

$$f_{vu}\left(F(\hat{\beta}_v - \hat{\beta}_u)\right) < f_{vu}\left(F(\tilde{\beta}_v - \tilde{\beta}_u)\right),$$

and this holds for every $(v,u) \in \mathcal{V} \times \mathcal{U}$. Hence, the log-likelihood at $\tilde{\beta}$ becomes

$$\begin{aligned}
\mathcal{L}(\tilde{\beta}) &= \sum_{\{x,y\} \subseteq \mathcal{U}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{\{x,y\} \subseteq \mathcal{V}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{(v,u) \in \mathcal{V} \times \mathcal{U}} f_{vu}\left(F(\tilde{\beta}_v - \tilde{\beta}_u)\right) \\
&> \sum_{\{x,y\} \subseteq \mathcal{U}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{\{x,y\} \subseteq \mathcal{V}} f_{xy}\left(F(\hat{\beta}_x - \hat{\beta}_y)\right) + \sum_{(v,u) \in \mathcal{V} \times \mathcal{U}} f_{vu}\left(F\left(\hat{\beta}_v - \hat{\beta}_u\right)\right) \\
&= \mathcal{L}(\hat{\beta}).
\end{aligned}$$

That is, $\mathcal{L}(\tilde{\beta}) > \mathcal{L}(\hat{\beta})$, leading to a contradiction. Hence, for every MLE $\hat{\beta}$, we must have $\|\hat{\beta}\|_\infty \leq |\mathcal{X}| \cdot \max_{(x,y)} \delta(x,y)$. □

E.2 Proof of Lemma 6.2.4

The log-likelihood function is given as

$$\mathcal{L}(\beta) = \sum_{(x,y) \in \mathcal{X}^2} \#\{x \succ y\} \log F(\beta_x - \beta_y).$$

Consider $\beta \neq \gamma \in \mathcal{D}$ and $\theta \in (0,1)$. Then,

$$\mathcal{L}(\theta\beta + (1-\theta)\gamma) = \sum_{(x,y)} \#\{x \succ y\} \log F(\theta\beta_x + (1-\theta)\gamma_x - \theta\beta_y - (1-\theta)\gamma_y)$$

$$
\begin{aligned}
&= \sum_{(x,y)} \#\{x \succ y\} \log F(\theta(\beta_x - \beta_y) + (1-\theta)(\gamma_x - \gamma_y)) \\
&\geq \sum_{(x,y)} \#\{x \succ y\} \big[\theta \log F(\beta_x - \beta_y) + (1-\theta) \log F(\gamma_x - \gamma_y)\big] \\
&= \theta \sum_{(x,y)} \#\{x \succ y\} \log F(\beta_x - \beta_y) + (1-\theta) \sum_{(x,y)} \#\{x \succ y\} \log F(\gamma_x - \gamma_y) \\
&= \theta \mathcal{L}(\beta) + (1-\theta)\mathcal{L}(\gamma),
\end{aligned}
$$

where the inequality holds because $\log F$ is concave, and $\#\{x \succ y\} \geq 0$ for every $(x, y) \in \mathcal{X}^2$. Hence, $\mathcal{L}$ is a concave function.

Next, suppose the comparison graph $\mathcal{G}_{\#}$ is connected. Recall, r denotes the reference alternative set to zero. As $\beta \neq \gamma$, this implies that there exists an alternative $a \neq r$ such that $\beta_a \neq \gamma_a$. We know that the graph $\mathcal{G}_{\#}$ is connected, hence, there exists an undirected path from a to r in $\mathcal{G}_{\#}$. Let this (undirected) path be given as

$$
a = v_0 \to v_1 \to v_2 \to \cdots \to v_t \to v_{t+1} = r.
$$

As $\beta_a - \beta_r \neq \gamma_a - \gamma_r$, this implies that there exists $(l, l+1)$ such that $\beta_{v_l} - \beta_{v_{l+1}} \neq \gamma_{v_l} - \gamma_{v_{l+1}}$. Because if this difference was equal for all $l \in [0, t]$, it would imply that $\beta_a - \beta_r = \gamma_a - \gamma_r$. As there's an edge between v_l and v_{l+1}, it implies that either $\#\{v_l \succ v_{l+1}\} > 0$ or $\#\{v_{l+1} \succ v_l\} > 0$. Without loss of generality, let $\#\{v_l \succ v_{l+1}\} > 0$. The log-likelihood is then

$$
\begin{aligned}
\mathcal{L}(\theta\beta + (1-\theta)\gamma) &= \#\{v_l \succ v_{l+1}\} \log F(\theta(\beta_{v_l} - \beta_{v_{l+1}}) + (1-\theta)(\gamma_{v_l} - \gamma_{v_{l+1}})) \\
&\qquad + \sum_{(x,y) \neq (v_l, v_{l+1})} \#\{x \succ y\} \log F(\theta(\beta_x - \beta_y) + (1-\theta)(\gamma_x - \gamma_y)) \\
&> \#\{v_l \succ v_{l+1}\} \left[\theta \log F(\beta_{v_l} - \beta_{v_{l+1}}) + (1-\theta) \log F(\gamma_{v_l} - \gamma_{v_{l+1}})\right] \\
&\qquad + \sum_{(x,y) \neq (v_l, v_{l+1})} \#\{x \succ y\} \left[\theta \log F(\beta_x - \beta_y) + (1-\theta) \log F(\gamma_x - \gamma_y)\right] \\
&= \theta \mathcal{L}(\beta) + (1-\theta)\mathcal{L}(\gamma)
\end{aligned}
$$

where the strict inequality holds because $\#\{v_l \succ v_{l+1}\} > 0, \theta \in (0, 1), \beta_{v_l} - \beta_{v_{l+1}} \neq \gamma_{v_l} - \gamma_{v_{l+1}}$ and $\log F$ is strictly concave. Therefore, $\mathcal{L}$ is strictly concave, and, it has unique maximizers.

For the converse, suppose the comparison graph $\mathcal{G}_{\#}$ is not connected (in the undirected form). As there is only one reference alternative r, let C be a connected component that does not contain r. The log-likelihood can then be rewritten as

$$
\mathcal{L}(\beta) = \sum_{x,y \in C} \#\{x \succ y\} \log F(\beta_x - \beta_y) + \sum_{x,y \notin C} \#\{x \succ y\} \log F(\beta_x - \beta_y),
$$

as there are no edges between C and its complement. Similar to proof of Lemma 6.2.1, for any vector $\beta \in \mathcal{D}$, define $\beta^{\Delta} \in \mathcal{D}$ for any $\Delta > 0$ as follows

$$
\beta_z^{\Delta} = \begin{cases} \beta_z + \Delta & ; \text{ if } z \in C \\ \beta_z & ; \text{ if } z \notin C. \end{cases}
$$

The likelihood at this point for any Δ is

$$\mathcal{L}(\beta^{\Delta}) = \sum_{x,y \in C} \#\{x \succ y\} \log F(\beta_x + \Delta - \beta_y - \Delta) + \sum_{x,y \notin C} \#\{x \succ y\} \log F(\beta_x - \beta_y) = \mathcal{L}(\beta). \tag{E.4}$$

Consider any $\theta \in (0,1)$. Then,

$$(\theta\beta^{\Delta} + (1-\theta)\beta)_z = \begin{cases} \theta(\beta_z + \Delta) + (1-\theta)\beta_z & = \beta_z + \theta\Delta \quad ; \text{ if } z \in C \\ \theta\beta_z + (1-\theta)\beta_z & = \beta_z \quad ; \text{ if } z \notin C, \end{cases}$$

and hence implying that $\theta\beta^{\Delta} + (1-\theta)\beta = \beta^{\theta\Delta}$. In particular, this gives us

$$\mathcal{L}(\theta\beta^{\Delta} + (1-\theta)\beta) = \mathcal{L}(\beta^{\theta\Delta}) = \mathcal{L}(\beta) = \theta\mathcal{L}(\beta^{\Delta}) + (1-\theta)\mathcal{L}(\beta),$$

where the second equality holds because Equation (E.4) holds for any $\Delta > 0$ (including $\theta\Delta$). But, as $\beta^{\Delta} \neq \beta$ and $\theta \in (0,1)$, this implies that $\mathcal{L}$ is not strictly concave. Note that, this also shows that if an MLE $\hat{\beta}$ existed, it would not be unique. As, $\hat{\beta}^{\Delta}$, with say $\Delta = 1$, would have the same likelihood as $\hat{\beta}$ making it an MLE as well.

Hence, concluding the proof that $\mathcal{L}(\beta)$ is strictly concave and the MLE is unique, iff the comparison graph $\mathcal{G}_{\#}$ is connected. □

E.3 Proof of Theorem 6.3.2

Suppose the dataset is such that it satisfies the properties given in Definition 6.3.1, i.e., $\#\{a \succ b\} > \#\{b \succ a\}$, and for every other alternative $x \in \mathcal{X} \setminus \{a,b\}$, we have

$$\#\{a \succ x\} > \#\{b \succ x\} \quad \text{and} \quad \#\{x \succ a\} < \#\{x \succ b\}.$$

Suppose for the sake of contradiction that there exists an MLE $\hat{\beta}$ such that $\hat{\beta}_a < \hat{\beta}_b$. Construct $\tilde{\beta}$ such that it is the same as $\hat{\beta}$, except with a's and b's utilities swapped.[6] That is,

$$\tilde{\beta}_x = \begin{cases} \hat{\beta}_x; \text{ if } x \notin \{a,b\} \\ \hat{\beta}_b; \text{ if } x = a \\ \hat{\beta}_a; \text{ if } x = b. \end{cases}$$

The log-likelihood at the MLE $\hat{\beta}$ is given as

$$\begin{aligned} \mathcal{L}(\beta) &= \sum_{(x,y) \in \mathcal{X}^2} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\ &= \sum_{x,y \notin \{a,b\}} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) \end{aligned}$$

[6] In case either a or b is the reference alternative, shift $\tilde{\beta}$ after swapping these two alternatives' utilities such that the reference is restored. Rest of the proof remains the same as the shifted beta vector has the same likelihood as the unshifted one.

$$
\begin{aligned}
&+ \sum_{y \notin \{a,b\}} \#\{a \succ y\} \log F(\hat{\beta}_a - \hat{\beta}_y) + \sum_{y \notin \{a,b\}} \#\{b \succ y\} \log F(\hat{\beta}_b - \hat{\beta}_y) \\
&+ \sum_{x \notin \{a,b\}} \#\{x \succ a\} \log F(\hat{\beta}_x - \hat{\beta}_a) + \sum_{x \notin \{a,b\}} \#\{x \succ b\} \log F(\hat{\beta}_x - \hat{\beta}_b) \\
&+ \#\{a \succ b\} \log F(\hat{\beta}_a - \hat{\beta}_b) + \#\{b \succ a\} \log F(\hat{\beta}_b - \hat{\beta}_a). \qquad \text{(E.5)}
\end{aligned}
$$

Before proceeding with the proof, we prove a simple claim.

Claim E.3.1. *Let $c, d, e, f > 0$ such that $c > d$ and $e > f$. Then $ce + df > cf + de$.*

Proof of Claim E.3.1.

$$
\begin{aligned}
ce + df &= c(f + (e - f)) + df \\
&= cf + c(e - f) + df \\
&> cf + d(e - f) + df \\
&= cf + de,
\end{aligned}
$$

where the inequality holds because $c > d$ and $(e - f) > 0$. □

By Claim E.3.1, for any $x, y \in \mathcal{X}$, we have,

$$
\begin{aligned}
&\#\{a \succ y\} \log F(\hat{\beta}_a - \hat{\beta}_y) + \#\{b \succ y\} \log F(\hat{\beta}_b - \hat{\beta}_y) \\
&\qquad < \#\{a \succ y\} \log F(\hat{\beta}_b - \hat{\beta}_y) + \#\{b \succ y\} \log F(\hat{\beta}_a - \hat{\beta}_y), \\
&\#\{x \succ a\} \log F(\hat{\beta}_x - \hat{\beta}_a) + \#\{x \succ b\} \log F(\hat{\beta}_x - \hat{\beta}_b) \\
&\qquad < \#\{x \succ b\} \log F(\hat{\beta}_x - \hat{\beta}_a) + \#\{x \succ a\} \log F(\hat{\beta}_x - \hat{\beta}_b), \\
&\#\{a \succ b\} \log F(\hat{\beta}_a - \hat{\beta}_b) + \#\{b \succ a\} \log F(\hat{\beta}_b - \hat{\beta}_a) \\
&\qquad < \#\{a \succ b\} \log F(\hat{\beta}_b - \hat{\beta}_a) + \#\{b \succ a\} \log F(\hat{\beta}_a - \hat{\beta}_b),
\end{aligned}
$$

using the property on the counts in the dataset, the fact that $\hat{\beta}_a < \hat{\beta}_b$ and F is strictly monotonic. Hence, using these expressions in Equation (E.5), we obtain

$$
\begin{aligned}
\mathcal{L}(\hat{\beta}) < &\sum_{x,y \notin \{a,b\}} \#\{x \succ y\} \log F(\hat{\beta}_x - \hat{\beta}_y) \\
&+ \sum_{y \notin \{a,b\}} \#\{a \succ y\} \log F(\hat{\beta}_b - \hat{\beta}_y) + \sum_{y \notin \{a,b\}} \#\{b \succ y\} \log F(\hat{\beta}_a - \hat{\beta}_y) \\
&+ \sum_{x \notin \{a,b\}} \#\{x \succ b\} \log F(\hat{\beta}_x - \hat{\beta}_a) + \sum_{x \notin \{a,b\}} \#\{x \succ a\} \log F(\hat{\beta}_x - \hat{\beta}_b) \\
&+ \#\{a \succ b\} \log F(\hat{\beta}_b - \hat{\beta}_a) + \#\{b \succ a\} \log F(\hat{\beta}_a - \hat{\beta}_b) \\
= &\sum_{x,y \notin \{a,b\}} \#\{x \succ y\} \log F(\tilde{\beta}_x - \tilde{\beta}_y) \\
&+ \sum_{y \notin \{a,b\}} \#\{a \succ y\} \log F(\tilde{\beta}_a - \tilde{\beta}_y) + \sum_{y \notin \{a,b\}} \#\{b \succ y\} \log F(\tilde{\beta}_b - \tilde{\beta}_y)
\end{aligned}
$$

$$
\begin{aligned}
&+ \sum_{x \notin \{a,b\}} \#\{x \succ b\} \log F(\tilde{\beta}_x - \tilde{\beta}_b) + \sum_{x \notin \{a,b\}} \#\{x \succ a\} \log F(\tilde{\beta}_x - \tilde{\beta}_a) \\
&+ \#\{a \succ b\} \log F(\tilde{\beta}_a - \tilde{\beta}_b) + \#\{b \succ a\} \log F(\tilde{\beta}_b - \tilde{\beta}_a) \\
&= \sum_{x \neq y} \#\{x \succ y\} \log F(\tilde{\beta}_x - \tilde{\beta}_y) \\
&= \mathcal{L}(\tilde{\beta}),
\end{aligned}
$$

implying that $\tilde{\beta}$ has a strictly higher log-likelihood than the MLE $\hat{\beta}$, leading to a contradiction. Therefore, every every MLE $\hat{\beta}$ must satisfy $\hat{\beta}_a \geq \hat{\beta}_b$ under this condition. □

E.4 Proof of Theorem 6.4.2

Let $\#$ and $\tilde{\#}$ be two datasets as defined in Definition 6.4.1, with (unique) MLEs $\hat{\beta}$ and $\tilde{\beta}$. That is, $\tilde{\#}$ is the same as $\#$, except with $\alpha > 0$ comparisons of $a \succ b$ added to it. We prove that for all alternatives $x \in \mathcal{X}$, we have

$$\tilde{\beta}_a - \tilde{\beta}_x \geq \hat{\beta}_a - \tilde{\beta}_x.$$

The proof for the b part ($\tilde{\beta}_b - \tilde{\beta}_x \leq \hat{\beta}_b - \tilde{\beta}_x$) is completely symmetric.

Let the log-likelihood function with respect to $\#$ be denoted by $\mathcal{L}$, while the log-likelihood function with respect to $\tilde{\#}$ be denoted by $\tilde{\mathcal{L}}$. Any alternative could be set as the reference, but we use a as the reference alternative in this proof for ease of exposition. As $\tilde{\#}$ is the same as $\#$, except with α additional $a \succ b$ comparisons, we have

$$\tilde{\mathcal{L}}(\beta) = \mathcal{L}(\beta) + \alpha \log F(\beta_a - \beta_b).$$

Let $\mathcal{U}$ denote the set of alternatives $u \in \mathcal{X} \setminus \{a\}$ for which $\tilde{\beta}_u - \tilde{\beta}_a \leq \hat{\beta}_u - \hat{\beta}_a$.[7] And, let $\mathcal{V}$ denote the set of alternatives $v \in \mathcal{X} \setminus \{a\}$ for which $\tilde{\beta}_v - \tilde{\beta}_a > \hat{\beta}_v - \hat{\beta}_a$. Our goal is to show that $\mathcal{U} = \mathcal{X} \setminus \{a\}$, or equivalently that $\mathcal{V} = \phi$.

First, we show that $b \in \mathcal{U}$. Suppose for the sake of contradiction, that $\tilde{\beta}_b - \tilde{\beta}_a > \hat{\beta}_b - \hat{\beta}_a$. Then, this implies that $\alpha \log F(\tilde{\beta}_a - \tilde{\beta}_b) < \alpha \log F(\hat{\beta}_a - \hat{\beta}_b)$ as both log and F are strictly monotonic, and $\alpha > 0$. Further, as $\tilde{\beta}$ maximizes $\mathcal{L}$, we have $\mathcal{L}(\tilde{\beta}) \leq \mathcal{L}(\hat{\beta})$. This implies that $\mathcal{L}(\tilde{\beta}) + \alpha \log F(\tilde{\beta}_a - \tilde{\beta}_b) < \mathcal{L}(\hat{\beta}) + \alpha \log F(\hat{\beta}_a - \hat{\beta}_b)$. Or, $\tilde{\mathcal{L}}(\tilde{\beta}) < \tilde{\mathcal{L}}(\hat{\beta})$, which is a contradiction as $\tilde{\beta}$ is the maximizer of $\tilde{\mathcal{L}}$. This proves that $\tilde{\beta}_b - \tilde{\beta}_a \leq \hat{\beta}_b - \hat{\beta}_a$, i.e. $b \in \mathcal{U}$.

Next, suppose for the sake of contradiction that $\mathcal{V} \neq \phi$. We can rewrite the log-likelihood function $\mathcal{L}(\beta)$ as

$$
\begin{aligned}
\mathcal{L}(\beta) &= \sum_{(x,y) \in \mathcal{X}^2} \#\{x \succ y\} \log F(\beta_x - \beta_y) \\
&= \sum_{\{x,y\} \subseteq \mathcal{X}} \Big[\#\{x \succ y\} \log F(\beta_x - \beta_y) + \#\{y \succ x\} \log F(\beta_y - \beta_x) \Big],
\end{aligned}
$$

[7] Even though $\tilde{\beta}_a = \hat{\beta}_a = 0$ as a is the reference, we do not omit it in some parts for better clarity.

where the latter summation is over unordered pairs of alternatives $\{x, y\}$ with $x \neq y$. Denote each term in this expression by $\ell_{xy}(\beta_x - \beta_y)$, i.e.

$$\ell_{xy}(\eta) = \#\{x \succ y\} \log F(\eta) + \#\{y \succ x\} \log F(-\eta).$$

As F is log-concave, $\log F$ is a concave function. And since linear transformations, positive scalar multiplication and addition preserve concavity, each of these functions ℓ_{xy} is also concave.

Let us define operator Δ_{xy} to be such that when it is applied to a β vector, it returns the difference in β values of alternatives x and y. That is, $\Delta_{xy}\beta := \beta_x - \beta_y$. Using this notation to rewrite the log-likelihood function, we have

$$\begin{aligned}\mathcal{L}(\beta) &= \sum_{\{x,y\}} \ell_{xy}(\Delta_{xy}\beta) \\ &= \sum_{u\in\mathcal{U}} \ell_{ua}(\Delta_{ua}\beta) + \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\beta) + \sum_{\{u,p\}\subseteq\mathcal{U}} \ell_{up}(\Delta_{up}\beta) \\ &\qquad + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\beta) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\beta),\end{aligned}$$

where the first two terms are the paired terms with a, the third is pairs within $\mathcal{U}$, the fourth is pairs within $\mathcal{V}$, and the last is for pairs across $\mathcal{U}$ and $\mathcal{V}$. For each $v \in \mathcal{V}$, we know $\tilde{\beta}_v - \tilde{\beta}_a > \hat{\beta}_v - \hat{\beta}_a$. Hence, we can write $\Delta_{va}\tilde{\beta} = \Delta_{va}\hat{\beta} + \delta_v$,[8] where $\delta_v > 0$ for each $v \in \mathcal{V}$. Recall, $\hat{\beta}$ is the maximizer of $\mathcal{L}$. Hence, $\mathcal{L}(\hat{\beta}_\mathcal{U}, \tilde{\beta}_\mathcal{V}) < \mathcal{L}(\hat{\beta}_\mathcal{U}, \hat{\beta}_\mathcal{V})$,[9] as the MLE $\hat{\beta}$ is unique, and $\mathcal{V} \neq \phi$. This implies that

$$\begin{aligned}&\sum_{u\in\mathcal{U}} \ell_{ua}(\Delta_{ua}\hat{\beta}) + \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\tilde{\beta}) + \sum_{\{u,p\}\subseteq\mathcal{U}} \ell_{up}(\Delta_{up}\hat{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\tilde{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\tilde{\beta}_v - \hat{\beta}_u) \\ <\ &\sum_{u\in\mathcal{U}} \ell_{ua}(\Delta_{ua}\hat{\beta}) + \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) + \sum_{\{u,p\}\subseteq\mathcal{U}} \ell_{up}(\Delta_{up}\hat{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\hat{\beta}_v - \hat{\beta}_u).\end{aligned}$$

Cancelling terms that appear on both sides (because of the same $\hat{\beta}_\mathcal{U}$), and plugging in $\Delta_{va}\tilde{\beta} = \Delta_{va}\hat{\beta} + \delta_v$, we have

$$\begin{aligned}&\sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta} + \delta_v) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta} + \delta_v - \delta_q) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \delta_v) \\ &\qquad < \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta}).\end{aligned}$$

Or in other words,

$$\begin{aligned}&\sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \delta_v) - \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta}) \qquad \text{(E.6)} \\ &\qquad < -\left[\sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta} + \delta_v) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta} + \delta_v - \delta_q) - \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) - \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta})\right].\end{aligned}$$

[8] Equivalently, this could be written as $\tilde{\beta}_v = \hat{\beta}_v + \delta_v$, as $\tilde{\beta}_a = \hat{\beta}_a = 0$.

[9] Recall that alternative a has been set as the reference, and hence it zero in both these terms.

Intuitively, it says that if you increase each $\hat{\beta}_v$ by their δ_v, the increase in likelihood because of the cross terms ℓ_{vu} is less than the loss because of the exclusive v terms (or vice versa, i.e. the loss in likelihood because of ℓ_{vu} is higher than the increase because of the exclusive v terms).

For each $u \in \mathcal{U}$, we know $\tilde{\beta}_u - \tilde{\beta}_a \leq \hat{\beta}_u - \hat{\beta}_a$. Hence, we can write $\Delta_{ua}\tilde{\beta} = \Delta_{ua}\hat{\beta} - \lambda_u$,[10] where $\lambda_u \geq 0$ for each $u \in \mathcal{U}$. We now compare $\tilde{\mathcal{L}}(\tilde{\beta}_{\mathcal{U}}, \tilde{\beta}_{\mathcal{V}})$ and $\tilde{\mathcal{L}}(\tilde{\beta}_{\mathcal{U}}, \hat{\beta}_{\mathcal{V}})$.[11] In other words, we compare

$$\sum_{u\in\mathcal{U}} \ell_{ua}(\Delta_{ua}\tilde{\beta}) + \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\tilde{\beta}) + \sum_{\{u,p\}\subseteq\mathcal{U}} \ell_{up}(\Delta_{up}\tilde{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\tilde{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\tilde{\beta}) + \alpha \log F(\Delta_{ab}\tilde{\beta})$$

$$vs$$

$$\sum_{u\in\mathcal{U}} \ell_{ua}(\Delta_{ua}\tilde{\beta}) + \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) + \sum_{\{u,p\}\subseteq\mathcal{U}} \ell_{up}(\Delta_{up}\tilde{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\hat{\beta}_v - \tilde{\beta}_u) + \alpha \log F(\Delta_{ab}\tilde{\beta}).$$

Note that the last term $\alpha \log F(\Delta_{ab}\beta)$ appears with a $\tilde{\beta}$ in both the equations because we know $b \in \mathcal{U}$. Cancelling terms that appear on both sides (because of the same $\tilde{\beta}_{\mathcal{U}}$), and plugging in $\Delta_{va}\tilde{\beta} = \Delta_{va}\hat{\beta} + \delta_v$ for $v \in \mathcal{V}$, as well as $\Delta_{ua}\tilde{\beta} = \Delta_{ua}\hat{\beta} - \lambda_u$ for $u \in \mathcal{U}$, we are comparing

$$\sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta} + \delta_v) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta} + \delta_v - \delta_q) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u + \delta_v)$$

$$vs$$

$$\sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta}) + \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u).$$

And, rearranging this, we compare

$$\sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u + \delta_v) - \sum_{(u,v)\in\mathcal{U}\times\mathcal{V}} \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u)$$

$$vs \tag{E.7}$$

$$-\left[\sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta} + \delta_v) + \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta} + \delta_v - \delta_q) - \sum_{v\in\mathcal{V}} \ell_{va}(\Delta_{va}\hat{\beta}) - \sum_{\{v,q\}\subseteq\mathcal{V}} \ell_{vq}(\Delta_{vq}\hat{\beta})\right].$$

If λ_u were zero, we know that the left hand side (i.e. the equation placed above in (E.7)) is smaller (than the one placed below) because of equation (E.6). But, we now show that

[10] Equivalently, this could be written as $\tilde{\beta}_u = \hat{\beta}_u - \lambda_u$, as $\tilde{\beta}_a = \hat{\beta}_a = 0$.

[11] That is, we are again keeping the $\mathcal{U}$ part fixed, while changing the $\mathcal{V}$ part from $\tilde{\beta}_{\mathcal{V}}$ to $\hat{\beta}_{\mathcal{V}}$.

this holds even for $\lambda_u \geq 0$ by concavity of the functions ℓ_{vu}. For each $(u, v) \in \mathcal{U} \times \mathcal{V}$, we can write

$$\ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u + \delta_v) - \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u) = \int_{\Delta_{vu}\hat{\beta}+\lambda_u}^{\Delta_{vu}\hat{\beta}+\lambda_u+\delta_v} \ell'_{vu}(t)dt,$$

where ℓ'_{vu} is the derivative of ℓ_{vu}.[12] Changing the variable of intergration,

$$\ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u + \delta_v) - \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u) = \int_{\Delta_{vu}\hat{\beta}}^{\Delta_{vu}\hat{\beta}+\delta_v} \ell'_{vu}(s + \lambda_u)ds.$$

But, we know that ℓ_{vu} is a concave function, implying that ℓ'_{vu} is monotonically decreasing. Hence, $\ell'_{vu}(s + \lambda_u) \leq \ell'_{vu}(s)$ for every s, as $\lambda_u \geq 0$. This gives us

$$\begin{aligned}
\ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u + \delta_v) - \ell_{vu}(\Delta_{vu}\hat{\beta} + \lambda_u) &= \int_{\Delta_{vu}\hat{\beta}}^{\Delta_{vu}\hat{\beta}+\delta_v} \ell'_{vu}(s + \lambda_u)ds \\
&\leq \int_{\Delta_{vu}\hat{\beta}}^{\Delta_{vu}\hat{\beta}+\delta_v} \ell'_{vu}(s)ds \\
&= \ell_{vu}(\Delta_{vu}\hat{\beta} + \delta_v) - \ell_{vu}(\Delta_{vu}\hat{\beta}).
\end{aligned}$$

Taking a summation of the left hand side over all $(u, v) \in \mathcal{U} \times \mathcal{V}$, shows that this summation is less than or equal to the left hand side of Equation (E.6). Hence, this summation is strictly smaller than the right hand side of Equation (E.6) (because of Equation (E.6) itself). This in turn implies that the equation placed above in (E.7) is strictly smaller than the one placed below. In other words, $\tilde{\mathcal{L}}(\tilde{\beta}) < \tilde{\mathcal{L}}(\tilde{\beta}_{\mathcal{U}}, \hat{\beta}_{\mathcal{V}})$, contradicting the fact that $\tilde{\beta}$ is the maximizer of $\tilde{\mathcal{L}}$. Hence, $V = \phi$. In other words, for each $x \in \mathcal{X} \setminus \{a\}$, $\tilde{\beta}_x - \tilde{\beta}_a \leq \hat{\beta}_x - \hat{\beta}_a$. □

E.5 Proof of Theorem 6.5.3

Consider $\mathcal{X} = \{a, b, c\}$, and let the dataset be as follows. $\#\{a \succ b\} = 5 + \epsilon, \#\{b \succ a\} = 5, \#\{a \succ c\} = 5+\epsilon, \#\{c \succ a\} = 5, \#\{b \succ c\} = 100$ and $\#\{c \succ b\} = 1$. Here, ϵ is a constant lying in $[0, 1]$. Observe that for any $\epsilon > 0$, this dataset conforms to Definition 6.5.1 if we label $x_1, x_2, x_3 = a, b, c$. To show violation of PMC, we show that there exists $\epsilon_o \in (0, 1]$ for which the (unique) MLE $\hat{\beta}$ violates the corresponding requirement of $\hat{\beta}_a \geq \hat{\beta}_b \geq \hat{\beta}_c$.

The log-likelihood function for this data is given by

$$\begin{aligned}
\mathcal{L}_\epsilon(\beta) = (5 + \epsilon) \log F(\beta_a - \beta_b) + 5 \log F(\beta_b - \beta_a) + 100 \log F(\beta_b - \beta_c) + \log F(\beta_c - \beta_b) \\
+ (5 + \epsilon) \log F(\beta_a - \beta_c) + 5 \log F(\beta_c - \beta_a).
\end{aligned}$$

Observe that every alternative has been compared with every other alternative, and hence, the comparison graph $\mathcal{G}_\#$ is strongly connected. Further, as F is strictly monotonic, continuous and strictly log-concave, the MLE exists and is unique for any $\epsilon \in [0, 1]$ (by Lemmas 6.2.1 and 6.2.4). Further, the log-likelihood $\mathcal{L}_\epsilon(\beta)$ is a strictly concave function (for each

[12] which exists, as F is differentiable.

$\epsilon \in [0,1]$). Any alternative could be set as the reference, but we use c as the reference alternative in this proof for ease of exposition. That is, our domain is $\mathcal{D} = \{\beta \in \mathbb{R}^{\mathcal{X}} : \beta_c = 0\}$. The (unique) maximum likelihood estimator is given by

$$\hat{\beta}(\epsilon) = \operatorname*{argmax}_{\beta \in \mathcal{D}} \mathcal{L}_\epsilon(\beta).$$

We first show that $\hat{\beta}(\epsilon)$ is a continuous function of ϵ. As F is strictly monotonic and continuous, for each $\epsilon \in [0,1]$, Lemma 6.2.3 tells us that the MLE is bounded as

$$\|\hat{\beta}(\epsilon)\|_\infty \leq |\mathcal{X}| \cdot \max_{(x,y)\in\mathcal{X}^2} \delta_\epsilon(x,y),$$

where δ_ϵ is the perfect-fit distance, but is now dependent on ϵ. For the dataset at hand, these perfect-fit distances are given by

$$\delta_\epsilon(a,b) = F^{-1}\left(\frac{5+\epsilon}{10+\epsilon}\right), \quad \delta_\epsilon(b,c) = F^{-1}\left(\frac{100}{101}\right) \quad \text{and} \quad \delta_\epsilon(a,c) = F^{-1}\left(\frac{5+\epsilon}{10+\epsilon}\right).$$

And, $\delta_\epsilon(b,a) = -\delta_\epsilon(a,b), \delta_\epsilon(c,b) = -\delta_\epsilon(b,c)$ and $\delta_\epsilon(c,a) = -\delta_\epsilon(a,c)$, as $F^{-1}(1-x) = -F^{-1}(x)$. Further, the first three distances are non-negative (making the remaining three non-positive) as $F^{-1}(x) \geq 0$ for $x \geq \frac{1}{2}$. Hence, the bound on the MLE simplifies to

$$\|\hat{\beta}(\epsilon)\|_\infty \leq 3 \cdot \max\left(F^{-1}\left(\frac{5+\epsilon}{10+\epsilon}\right), F^{-1}\left(\frac{100}{101}\right)\right).$$

As F is strictly monotonic, it implies that F^{-1} is also strictly increasing. Applying this, we have

$$F^{-1}\left(\frac{5+\epsilon}{10+\epsilon}\right) \leq F^{-1}\left(\frac{6}{11}\right) < F^{-1}\left(\frac{100}{101}\right),$$

as $\epsilon \in [0,1]$. Therefore, the bound on the MLE further simplifies to

$$\|\hat{\beta}(\epsilon)\|_\infty \leq 3\ F^{-1}\left(\frac{100}{101}\right),$$

for any $\epsilon \in [0,1]$. Hence, the MLE optimization problem can be rewritten as

$$\hat{\beta}(\epsilon) = \operatorname*{argmax}_{\beta\in\mathcal{D}:\|\beta\|_\infty \leq 3\ F^{-1}\left(\frac{100}{101}\right)} \mathcal{L}_\epsilon(\beta).$$

This shows that we are optimizing over a compact space. Hence, by the Theorem of the Maximum [Ber63; JR11], both the maximum likelihood and the corresponding maximizer $\hat{\beta}(\epsilon)$ are continuous in the parameter ϵ, for all $\epsilon \in [0,1]$.

Next, we analyze the MLE at $\epsilon = 0$. The log-likelihood function for this value of ϵ is

$$\begin{aligned}\mathcal{L}_0(\beta) = {} & 5\log F(\beta_a - \beta_b) + 5\log F(\beta_b - \beta_a) + 100\log F(\beta_b - \beta_c) + \log F(\beta_c - \beta_b) \\ & + 5\log F(\beta_a - \beta_c) + 5\log F(\beta_c - \beta_a). \end{aligned} \tag{E.8}$$

For ease of exposition, we use $\beta^\dagger$ to denote the MLE when $\epsilon = 0$, i.e.

$$\beta^\dagger := \hat{\beta}(0) = \operatorname*{argmax}_{\beta \in \mathcal{D}} \mathcal{L}_0(\beta).$$

Recall that we used c as the reference alternative, and hence, $\beta_c^\dagger = 0$. Our goal is to show that $\beta_b^\dagger > \beta_a^\dagger > \beta_c^\dagger$. To this end, we first show that $\beta_a^\dagger = \beta_b^\dagger/2$, i.e. in terms of β values, a lies at the mid-point of b and c. Suppose for the sake of contradiction that $\beta_a^\dagger \neq \beta_b^\dagger/2$. Consider another vector $\tilde{\beta} \in \mathcal{D}$ that is the same as $\beta^\dagger$, except with the distances between β values of b & a and a & c swapped. This can be achieved by setting

$$\tilde{\beta}_x = \begin{cases} \beta_x^\dagger & ; \text{ if } x \in \{b, c\} \\ \beta_b^\dagger - \beta_a^\dagger & ; \text{ if } x = a. \end{cases}$$

Then, we have $\tilde{\beta}_a - \tilde{\beta}_c = \beta_b^\dagger - \beta_a^\dagger$ and $\tilde{\beta}_b - \tilde{\beta}_a = \beta_a^\dagger - \beta_c^\dagger$. Hence, the log-likelihood at this point is given by

$$\begin{aligned} \mathcal{L}_0(\tilde{\beta}) &= 5 \log F(\tilde{\beta}_a - \tilde{\beta}_b) + 5 \log F(\tilde{\beta}_b - \tilde{\beta}_a) + 100 \log F(\tilde{\beta}_b - \tilde{\beta}_c) + \log F(\tilde{\beta}_c - \tilde{\beta}_b) \\ &\quad + 5 \log F(\tilde{\beta}_a - \tilde{\beta}_c) + 5 \log F(\tilde{\beta}_c - \tilde{\beta}_a) \\ &= 5 \log F(\beta_c^\dagger - \beta_a^\dagger) + 5 \log F(\beta_a^\dagger - \beta_c^\dagger) + 100 \log F(\beta_b^\dagger - \beta_c^\dagger) + \log F(\beta_c^\dagger - \beta_b^\dagger) \\ &\quad + 5 \log F(\beta_b^\dagger - \beta_a^\dagger) + 5 \log F(\beta_a^\dagger - \beta_b^\dagger) \\ &= \mathcal{L}_0(\beta^\dagger). \end{aligned}$$

That is, swapping these distances does not change the likelihood, because of symmetry. Now, consider a new vector $\bar{\beta} = (\beta^\dagger + \tilde{\beta})/2$. Note that, as $\beta_a^\dagger \neq \beta_b^\dagger/2$, it implies that $\beta_a^\dagger \neq \beta_b^\dagger - \beta_a^\dagger = \tilde{\beta}_a$. In other words, $\tilde{\beta} \neq \beta^\dagger$. Therefore, applying strict concavity of $\mathcal{L}_0$, we have

$$\mathcal{L}_0(\bar{\beta}) = \mathcal{L}_0\left(\frac{\beta^\dagger + \tilde{\beta}}{2}\right) > \frac{\mathcal{L}_0(\beta^\dagger) + \mathcal{L}_0(\tilde{\beta})}{2} = \mathcal{L}_0(\beta^\dagger),$$

which is a contradiction as $\beta^\dagger$ is the maximizer of $\mathcal{L}_0$. This proves that $\beta_a^\dagger = \beta_b^\dagger/2$. In other words, $\beta^\dagger$ is of the form $(\beta_b^\dagger/2, \beta_b^\dagger, 0)$. Hence, $\beta^\dagger$ continues to be the maximizer of $\mathcal{L}_0$ among the vectors $\mathcal{A} = \{(\alpha/2, \alpha, 0) : \alpha \in \mathbb{R}\} \subseteq \mathcal{D}$. Rewriting the log-likelihood (E.8) for vectors in $\mathcal{A}$, we have

$$\begin{aligned} \mathcal{L}_0((\alpha/2, \alpha, 0)) &= 5 \log F\left(-\frac{\alpha}{2}\right) + 5 \log F\left(\frac{\alpha}{2}\right) + 100 \log F(\alpha) + \log F(-\alpha) \\ &\quad + 5 \log F\left(\frac{\alpha}{2}\right) + 5 \log F\left(-\frac{\alpha}{2}\right) \\ &= 10 \log F\left(\frac{\alpha}{2}\right) + 10 \log F\left(-\frac{\alpha}{2}\right) + 100 \log F(\alpha) + \log F(-\alpha). \end{aligned}$$

Overloading notation, we denote this log-likelihood by $\mathcal{L}_0(\alpha)$, and this is maximized at $\alpha = \beta_b^\dagger$. For ease of exposition, denote the composition of log and F by G, i.e. $G := \log F$. As

F is strictly monotonic, differentiable and strictly log-concave, G is also strictly monotonic and differentiable, and is strictly concave.[13] Rewriting the log-likelihood with this notation, we have

$$\mathcal{L}_0(\alpha) = 10G\left(\frac{\alpha}{2}\right) + 10G\left(-\frac{\alpha}{2}\right) + 100G\left(\alpha\right) + G\left(-\alpha\right).$$

We show that this function is not maximized at any $\alpha \leq 0$. In other words, $\beta_b^\dagger > 0$. Computing the derivative of $\mathcal{L}_0$, we have

$$\mathcal{L}_0'(\alpha) = 5G'\left(\frac{\alpha}{2}\right) - 5G'\left(-\frac{\alpha}{2}\right) + 100G'\left(\alpha\right) - G'\left(-\alpha\right).$$

As G is strictly concave, it implies that G' is strictly decreasing. Hence, for $\alpha \leq 0$, it implies that $G'(\frac{\alpha}{2}) \geq G'(-\frac{\alpha}{2})$ and $G'(\alpha) \geq G'(-\alpha)$. This shows that for $\alpha \leq 0$, we have

$$\mathcal{L}_0'(\alpha) \geq 99G'\left(\alpha\right) > 0,$$

where the last inequality holds as G is a strictly increasing function, leading to G' being positive.[14] In other words, if $\alpha \leq 0$, the log-likelihood can be strictly increased by taking an infinitesimally small step in the direction $\left[\frac{1}{2}, 1, 0\right]$. Hence, none of these points maximizes $\mathcal{L}_0$, and $\beta_b^\dagger > 0$. Also, recall that $\beta^\dagger$ was of the form $(\beta_b^\dagger/2, \beta_b^\dagger, 0)$; this proves that $\beta_b^\dagger > \beta_a^\dagger > \beta_c^\dagger$.

Finally, recall that we need $\epsilon > 0$ for the dataset to conform to Definition 6.5.1 with the labelling $x_1, x_2, x_3, = a, b, c$. To be able to find such a value of ϵ, we use continuity of $\hat{\beta}(\epsilon)$. By continuity, we know that for every $\gamma > 0$, there exists $\delta > 0$ such that $\|\hat{\beta}(\epsilon) - \hat{\beta}(0)\|_\infty < \gamma$ for all $|\epsilon - 0| < \delta$. Define $\theta := \beta_b^\dagger - \beta_a^\dagger > 0$. Then, choose $\gamma = \theta/3$, and let δ_o denote the corresponding value of δ. Hence, choose $\epsilon_o = \min(\delta_o/2, 1) > 0$. For this value of ϵ_o, we indeed have $\|\hat{\beta}(\epsilon_o) - \hat{\beta}(0)\|_\infty < \theta/3$. That is,

$$\hat{\beta}(\epsilon_o)_b > \beta_b^\dagger - \frac{\theta}{3} \qquad \text{and} \qquad \hat{\beta}(\epsilon_o)_a < \beta_a^\dagger + \frac{\theta}{3}.$$

Hence,

$$\hat{\beta}(\epsilon_o)_b - \hat{\beta}(\epsilon_o)_a > \beta_b^\dagger - \beta_a^\dagger - \frac{2\theta}{3} = \theta - \frac{2\theta}{3} > 0.$$

Therefore, at $\epsilon = \epsilon_o \in (0, 1]$, the MLE satisfies $\hat{\beta}(\epsilon_o)_b > \hat{\beta}(\epsilon_o)_a$. Hence, the dataset with $\epsilon = \epsilon_o$ satisfies the PMC condition, but the corresponding MLE does not conform to the corresponding ordering, proving violation of pairwise majority consistency. □

[13] This part of the proof does not require concavity of G, but we use it nevertheless as it simplifies the proof.

[14] Strictly speaking, a function might be strictly increasing and have a derivative that is not strictly positive at every point (in particular, the derivative might be zero at stationary points). But in our case, as G' is also a strictly decreasing function, it cannot be zero at any point, because that would make it negative at larger points, violating strict monotonicity of G.

E.6 Proof of Theorem 6.6.3

Consider $\mathcal{X} = \{a, b, c\}$, and let the two datasets be as follows. The first dataset is such that $\#^1\{a \succ c\} = 5 + \epsilon, \#^1\{c \succ a\} = 5 - \epsilon, \#^1\{c \succ b\} = 100, \#^1\{b \succ c\} = 1$, and has zero counts otherwise. The second dataset is such that $\#^2\{a \succ c\} = 5 + \epsilon, \#^2\{c \succ a\} = 5 - \epsilon, \#^2\{b \succ a\} = 100, \#^2\{a \succ b\} = 1$, and has zero counts otherwise. Here, ϵ is a constant lying in $(0, 1]$.

First, we analyze the MLE for the dataset $\#^1$. As the comparison graph $\mathcal{G}_{\#^1}$ is strongly connected, and F is strictly monotonic, continuous and strictly log-concave, the MLE $\hat{\beta}^1$ exists and is unique (by Lemmas 6.2.1 and 6.2.4). Further, the pair (a, c) satisfies the condition of Lemma 6.2.2, similarly does the pair (b, c). Applying the lemma for the pair (a, c) says that the MLE satisfies

$$\hat{\beta}^1_a = \hat{\beta}^1_c + F^{-1}\left(\frac{5+\epsilon}{10}\right) > \hat{\beta}^1_c,$$

as $(5 + \epsilon)/10$ is larger than $1/2$. Similarly, applying Lemma 6.2.2 for the pair (b, c) says that the MLE satisfies

$$\hat{\beta}^1_b = \hat{\beta}^1_c + F^{-1}\left(\frac{1}{1+100}\right) < \hat{\beta}^1_c,$$

as $1/101$ is smaller than $1/2$. Putting these equations together, we have $\hat{\beta}^1_a > \hat{\beta}^1_c > \hat{\beta}^1_b$.

Next, we analyze the MLE for the dataset $\#^2$. As the comparison graph $\mathcal{G}_{\#^2}$ is strongly connected, and F is strictly monotonic, continuous and strictly log-concave, the MLE $\hat{\beta}^2$ exists and is unique (by Lemmas 6.2.1 and 6.2.4). Further, the pair (c, a) satisfies the condition of Lemma 6.2.2, similarly does the pair (b, a). Applying the lemma for the pair (c, a) says that the MLE satisfies

$$\hat{\beta}^2_c = \hat{\beta}^2_a + F^{-1}\left(\frac{5-\epsilon}{10}\right) < \hat{\beta}^2_a,$$

as $(5 - \epsilon)/10$ is smaller than $1/2$. Similarly, applying Lemma 6.2.2 for the pair (b, a) says that the MLE satisfies

$$\hat{\beta}^2_b = \hat{\beta}^2_a + F^{-1}\left(\frac{100}{1+100}\right) > \hat{\beta}^2_a,$$

as $100/101$ is larger than $1/2$. Putting these equations together, we have $\hat{\beta}^2_b > \hat{\beta}^2_a > \hat{\beta}^2_c$. Hence, both datasets $\#^1$ and $\#^2$ have MLEs $\hat{\beta}^1$ and $\hat{\beta}^2$ such that $\hat{\beta}^1_a > \hat{\beta}^1_c$ and $\hat{\beta}^2_a > \hat{\beta}^2_c$.

Finally, we analyze the MLE for the dataset $\# = \#^1 + \#^2$ obtained by pooling both datasets $\#^1$ and $\#^2$. Recall that the proof so far holds for any constant $\epsilon \in (0, 1]$; but, from this point on, we allow ϵ to take the value of zero as well, i.e. $\epsilon \in [0, 1]$. The log-likelihood function for the pooled data $\#$ is given by

$$\begin{aligned}\mathcal{L}_\epsilon(\beta) = {} & 100 \log F(\beta_c - \beta_b) + \log F(\beta_b - \beta_c) + 100 \log F(\beta_b - \beta_a) + \log F(\beta_a - \beta_b) \\ & + (10 + 2\epsilon) \log F(\beta_a - \beta_c) + (10 - 2\epsilon) \log F(\beta_c - \beta_a).\end{aligned}$$

Observe that every alternative has been compared with every other alternative, and hence, the comparison graph $\mathcal{G}_{\#}$ is strongly connected. Further, as F is strictly monotonic, continuous and strictly log-concave, the MLE exists and is unique for any $\epsilon \in [0,1]$ (by Lemmas 6.2.1 and 6.2.4). Further, the log-likelihood $\mathcal{L}_\epsilon(\beta)$ is a strictly concave function (for each $\epsilon \in [0,1]$). Any alternative could be set as the reference, but we use a as the reference alternative in this proof for ease of exposition. That is, our domain is $\mathcal{D} = \{\beta \in \mathbb{R}^{\mathcal{X}} : \beta_a = 0\}$. The (unique) maximum likelihood estimator is given by

$$\hat{\beta}(\epsilon) = \operatorname*{argmax}_{\beta \in \mathcal{D}} \mathcal{L}_\epsilon(\beta).$$

Similar to the proof of Theorem 6.5.3, we first show that $\hat{\beta}(\epsilon)$ is a continuous function of ϵ. As F is strictly monotonic and continuous, for each $\epsilon \in [0,1]$, Lemma 6.2.3 tells us that the MLE is bounded as

$$\|\hat{\beta}(\epsilon)\|_\infty \leq |\mathcal{X}| \cdot \max_{(x,y)\in\mathcal{X}^2} \delta_\epsilon(x,y),$$

where δ_ϵ is the perfect-fit distance, but is now dependent on ϵ. For our pooled dataset, these perfect-fit distances are given by

$$\delta_\epsilon(b,a) = F^{-1}\left(\frac{100}{101}\right), \quad \delta_\epsilon(c,b) = F^{-1}\left(\frac{100}{101}\right) \quad \text{and} \quad \delta_\epsilon(a,c) = F^{-1}\left(\frac{10+2\epsilon}{20}\right).$$

And, $\delta_\epsilon(a,b) = -\delta_\epsilon(b,a), \delta_\epsilon(b,c) = -\delta_\epsilon(c,b)$ and $\delta_\epsilon(c,a) = -\delta_\epsilon(a,c)$, as $F^{-1}(1-x) = -F^{-1}(x)$. Further, the first three distances are non-negative (making the remaining three non-positive) as $F^{-1}(x) \geq 0$ for $x \geq \frac{1}{2}$. Hence, the bound on the MLE simplifies to

$$\|\hat{\beta}(\epsilon)\|_\infty \leq 3 \cdot \max\left(F^{-1}\left(\frac{100}{101}\right), F^{-1}\left(\frac{10+2\epsilon}{20}\right)\right).$$

As F is strictly monotonic, it implies that F^{-1} is also strictly increasing. Applying this, we have

$$F^{-1}\left(\frac{10+2\epsilon}{20}\right) \leq F^{-1}\left(\frac{12}{20}\right) < F^{-1}\left(\frac{100}{101}\right),$$

as $\epsilon \in [0,1]$. Therefore, the bound on the MLE further simplifies to

$$\|\hat{\beta}(\epsilon)\|_\infty \leq 3\ F^{-1}\left(\frac{100}{101}\right),$$

for any $\epsilon \in [0,1]$. Hence, the MLE optimization problem can be rewritten as

$$\hat{\beta}(\epsilon) = \operatorname*{argmax}_{\beta \in \mathcal{D}: \|\beta\|_\infty \leq 3\ F^{-1}\left(\frac{100}{101}\right)} \mathcal{L}_\epsilon(\beta).$$

This shows that we are optimizing over a compact space. Hence, by the Theorem of the Maximum, both the maximum likelihood and the corresponding maximizer $\hat{\beta}(\epsilon)$ are continuous in the parameter ϵ, for all $\epsilon \in [0,1]$.

Next, we analyze the MLE at $\epsilon = 0$. The log-likelihood function for this value of ϵ is

$$\begin{aligned}\mathcal{L}_0(\beta) = 100 \log F(\beta_c - \beta_b) + \log F(\beta_b - \beta_c) + 100 \log F(\beta_b - \beta_a) + \log F(\beta_a - \beta_b) \\ + 10 \log F(\beta_a - \beta_c) + 10 \log F(\beta_c - \beta_a). \end{aligned} \tag{E.9}$$

For ease of exposition, we use $\beta^\dagger$ to denote the MLE when $\epsilon = 0$, i.e.

$$\beta^\dagger := \hat{\beta}(0) = \operatorname*{argmax}_{\beta \in \mathcal{D}} \mathcal{L}_\epsilon(\beta).$$

Recall that we used a as the reference alternative, and hence, $\beta^\dagger_a = 0$. Our goal is to show that $\beta^\dagger_c > \beta^\dagger_b > \beta^\dagger_a$. To this end, we first show that $\beta^\dagger_b = \beta^\dagger_c/2$, i.e. in terms of β values, b lies at the mid-point of c and a. Suppose for the sake of contradiction that $\beta^\dagger_b \neq \beta^\dagger_c/2$. Consider another vector $\tilde{\beta} \in \mathcal{D}$ that is the same as $\beta^\dagger$, except with the distances between c & b and b & a swapped. This can be achieved by setting

$$\tilde{\beta}_x = \begin{cases} \beta^\dagger_x & ;\ \text{if } x \neq \{a, c\} \\ \beta^\dagger_c - \beta^\dagger_b & ;\ \text{if } x = b. \end{cases}$$

Then, we have $\tilde{\beta}_b - \tilde{\beta}_a = \beta^\dagger_c - \beta^\dagger_b$ and $\tilde{\beta}_c - \tilde{\beta}_b = \beta^\dagger_b - \beta^\dagger_a$. Hence, the likelihood at this point is given by

$$\begin{aligned}\mathcal{L}_0(\tilde{\beta}) &= 100 \log F(\tilde{\beta}_c - \tilde{\beta}_b) + \log F(\tilde{\beta}_b - \tilde{\beta}_c) + 100 \log F(\tilde{\beta}_b - \tilde{\beta}_a) + \log F(\tilde{\beta}_a - \tilde{\beta}_b) \\ &\quad + 10 \log F(\tilde{\beta}_a - \tilde{\beta}_c) + 10 \log F(\tilde{\beta}_c - \tilde{\beta}_a) \\ &= 100 \log F(\beta^\dagger_b - \beta^\dagger_a) + \log F(\beta^\dagger_a - \beta^\dagger_b) + 100 \log F(\beta^\dagger_c - \beta^\dagger_b) + \log F(\beta^\dagger_b - \beta^\dagger_c) \\ &\quad + 10 \log F(\beta^\dagger_a - \beta^\dagger_c) + 10 \log F(\beta^\dagger_c - \beta^\dagger_a) \\ &= \mathcal{L}_0(\beta^\dagger).\end{aligned}$$

That is, swapping these distances does not change the likelihood, because of symmetry. Now, consider a new vector $\bar{\beta} = (\beta^\dagger + \tilde{\beta})/2$. Note that, as $\beta^\dagger_b \neq \beta^\dagger_c/2$, it implies that $\beta^\dagger_b \neq \beta^\dagger_c - \beta^\dagger_b = \tilde{\beta}_b$. In other words, $\tilde{\beta} \neq \beta^\dagger$. Therefore, applying strict concavity of $\mathcal{L}_0$, we have

$$\mathcal{L}_0(\bar{\beta}) = \mathcal{L}_0\left(\frac{\beta^\dagger + \tilde{\beta}}{2}\right) > \frac{\mathcal{L}_0(\beta^\dagger) + \mathcal{L}_0(\tilde{\beta})}{2} = \mathcal{L}_0(\beta^\dagger),$$

which is a contradiction as $\beta^\dagger$ is the maximizer of $\mathcal{L}_0$. This proves that $\beta^\dagger_b = \beta^\dagger_c/2$. In other words, $\beta^\dagger$ is of the form $(0, \beta^\dagger_c/2, \beta^\dagger_c)$. Hence, $\beta^\dagger$ continues to be the maximizer of $\mathcal{L}_0$ among the vectors $\mathcal{A} = \{(0, \alpha/2, \alpha) : \alpha \in \mathbb{R}\} \subseteq \mathcal{D}$. Rewriting the log-likelihood (E.9) for vectors in $\mathcal{A}$, we have

$$\begin{aligned}\mathcal{L}_0((0, \alpha/2, \alpha)) = 100 \log F\left(\frac{\alpha}{2}\right) + \log F\left(-\frac{\alpha}{2}\right) + 100 \log F\left(\frac{\alpha}{2}\right) + \log F\left(-\frac{\alpha}{2}\right) \\ + 10 \log F\left(-\alpha\right) + 10 \log F\left(\alpha\right)\end{aligned}$$

$$= 200 \log F\left(\frac{\alpha}{2}\right) + 2 \log F\left(-\frac{\alpha}{2}\right) + 10 \log F(\alpha) + 10 \log F(-\alpha).$$

Overloading notation, we denote this log-likelihood by $\mathcal{L}_0(\alpha)$, and this is maximized at $\alpha = \beta_c^\dagger$. For ease of exposition, denote the composition of log and F by G, i.e. $G := \log F$. As F is strictly monotonic, differentiable and strictly log-concave, G is also strictly monotonic and differentiable, and is strictly concave.[15] Rewriting the log-likelihood with this notation, we have

$$\mathcal{L}_0(\alpha) = 200G\left(\frac{\alpha}{2}\right) + 2G\left(-\frac{\alpha}{2}\right) + 10G(\alpha) + 10G(-\alpha).$$

We show that this function is not maximized at any $\alpha \leq 0$. In other words, $\beta_c^\dagger > 0$. Computing the derivative of $\mathcal{L}_0$, we have

$$\mathcal{L}_0'(\alpha) = 100G'\left(\frac{\alpha}{2}\right) - G'\left(-\frac{\alpha}{2}\right) + 10G'(\alpha) - 10G'(-\alpha).$$

As G is strictly concave, it implies that G' is strictly decreasing. Hence, for $\alpha \leq 0$, it implies that $G'(\frac{\alpha}{2}) \geq G'(-\frac{\alpha}{2})$ and $G'(\alpha) \geq G'(-\alpha)$. This shows that for $\alpha \leq 0$, we have

$$\mathcal{L}_0'(\alpha) \geq 99G'\left(\frac{\alpha}{2}\right) > 0,$$

where the last inequality holds as G is a strictly increasing function, leading to G' being positive.[16] In other words, if $\alpha \leq 0$, the log-likelihood can be strictly increased by taking an infinitesimally small step in the direction $\left[0, \frac{1}{2}, 1\right]$. Hence, none of these points maximizes $\mathcal{L}_0$, and $\beta_c^\dagger > 0$. Also, recall that $\beta^\dagger$ was of the form $(0, \beta_c^\dagger/2, \beta_c^\dagger)$; this proves that $\beta_c^\dagger > \beta_b^\dagger > \beta_a^\dagger$.

Finally, recall that the initial part of the proof (analyzing the MLE for the individual datasets) works only for $0 < \epsilon \leq 1$. Hence, we need to use an ϵ value strictly larger than zero even for the pooled dataset. To be able to find such a value of ϵ, we use continuity of $\hat{\beta}(\epsilon)$. By continuity, we know that for every $\gamma > 0$, there exists $\delta > 0$ such that $\|\hat{\beta}(\epsilon) - \hat{\beta}(0)\|_\infty < \gamma$ for all $|\epsilon - 0| < \delta$. Define $\theta := \beta_c^\dagger - \beta_a^\dagger > 0$. Then, choose $\gamma = \theta/3$, and let δ_o denote the corresponding value of δ. Hence, choose $\epsilon_o = \min(\delta_o/2, 1) > 0$. For this value of ϵ_o, we indeed have $\|\hat{\beta}(\epsilon_o) - \hat{\beta}(0)\|_\infty < \theta/3$. That is,

$$\hat{\beta}(\epsilon_o)_c > \beta_c^\dagger - \frac{\theta}{3} \qquad \text{and} \qquad \hat{\beta}(\epsilon_o)_a < \beta_a^\dagger + \frac{\theta}{3}.$$

Hence,

$$\hat{\beta}(\epsilon_o)_c - \hat{\beta}(\epsilon_o)_a > \beta_c^\dagger - \beta_a^\dagger - \frac{2\theta}{3} = \theta - \frac{2\theta}{3} > 0.$$

Therefore, at $\epsilon = \epsilon_o \in (0, 1]$, the MLE (on the pooled data) satisfies $\hat{\beta}(\epsilon_o)_c > \hat{\beta}(\epsilon_o)_a$. Hence, for $\epsilon = \epsilon_o$, the two datasets $\#^1$ and $\#^2$ have MLEs $\hat{\beta}^1$ and $\hat{\beta}^2$ such that $\hat{\beta}_a^1 > \hat{\beta}_c^1$ and $\hat{\beta}_a^2 > \hat{\beta}_c^2$, but the MLE $\hat{\beta}$ on the pooled dataset $\# = \#^1 + \#^2$ satisfies $\hat{\beta}_a < \hat{\beta}_c$, proving violation of separability. □

[15] This part of the proof does not require concavity of G, but we use it nevertheless as it simplifies the proof.

[16] Strictly speaking, a function might be strictly increasing and have a derivative that is not strictly positive at every point (in particular, the derivative might be zero at stationary points). But in our case, as G' is also a strictly decreasing function, it cannot be zero at any point, because that would make it negative at larger points, violating strict monotonicity of G.

Appendix F

Omitted Proofs for Chapter 7

F.1 Proof of Lemma 7.3.2

Consider an instance of the CAIM problem with prior probability distribution $\mathbf{\Phi}$ that is the realization ϕ_* with probability 1, where ϕ_* is a vector of all 1s. Such a problem reduces to the standard influence maximization problem, wherein we need to find the optimal subset of K nodes to influence to have maximum influence spread in the network. But, the standard influence maximization problem is an NP-Hard problem, making CAIM NP-Hard too.

F.2 Proof of Lemma 7.3.3

The key idea is that taking a particular action (say a_o) now, may have a low marginal gain because of the realization of the current session, but after a few actions, taking the same action a_o might have a high marginal gain because of a change of session.

More formally, consider the following example. At the beginning of the first session, we take a query action and ask about nodes $\{1, 2, 3\}$. We get the observation that each of them is absent. At this point, if we take the invite action $a_o = \langle\{2\}, i\rangle$, we get a marginal gain of 0. On the other hand, suppose we took the end-session action after the query, advance to the next session, again take a query action and ask about nodes $\{1, 2, 3\}$ and this time get the observation that 2 is present (while others are absent). Now if we take the same invite action a_o, we get a positive marginal gain. This shows that the objective function of CAIM is not adaptive submodular.

F.3 Proof of Theorem 7.5.1

The difference between $\sum_{x=1}^{\ell} \mathcal{I}(S_{P_x})$ and $\mathcal{I}(\boldsymbol{S})$ comes from the fact that $\sum_{x=1}^{\ell} \mathcal{I}(S_{P_x})$ over-counts influence spread across communities [since $\mathcal{I}(S_{P_x})$ equals the expected influence in the whole graph when S_{P_x} is influenced, assuming no nodes of other communities are influenced, while in fact some actually may be].

Edges going across communities lead to this double counting of influence spread. We'll call these edges as cross-edges. Let M_a denote the total number of such cross-edges, i.e. $M_a = |\{(u,v) \in E : u \in P_x, v \in P_y \text{ and } x \neq y\}|$. Each cross-edge can lead to at most two nodes being double counted. This is because of the following: Let (u,v) be a cross-edge (where $u \in P_x$ and $v \in P_y$), and suppose that both these nodes are influenced. On computing $\mathcal{I}(S_{P_x})$, v might be counted as being influenced by it [even though v is already influenced beforehand], hence leading to an over-count of 1 [Note that, since we're considering one round of influence spread, $\mathcal{I}(S_{P_x})$ assumes that v does not propagate influence further]. Similar holds with $\mathcal{I}(S_{P_y})$.

Hence, $\sum_{x=1}^{\ell} \mathcal{I}(S_{P_x}) - \mathcal{I}(\boldsymbol{S})$ is bounded by twice the expected number of cross-edges that are activated (for arbitrary $\boldsymbol{S}$). Let E_{ij} be the random variable denoting whether there's an edge from node i to j in the SBM network. Then, the number of cross-edges is given as

$$M_a = \frac{1}{2} \sum_{x=1}^{\ell} \sum_{i \in P_x} \sum_{j \notin P_x} E_{ij},$$

Hence, the expected number of cross edges is

$$\begin{aligned}
\mathbb{E}[M_a] &= \mathbb{E}\left[\frac{1}{2} \sum_{x=1}^{\ell} \sum_{i \in P_x} \sum_{j \notin P_x} E_{ij}\right] \\
&= \frac{1}{2} \sum_{x=1}^{\ell} \sum_{i \in P_x} \sum_{j \notin P_x} q = \frac{1}{2} \sum_{x=1}^{\ell} \sum_{i \in P_x} (n - |P_x|)q \\
&= \frac{q}{2} \sum_{x=1}^{\ell} |P_x|(n - |P_x|) \\
&= \frac{q}{2} \left(n^2 - \sum_{x=1}^{\ell} |P_x|^2\right).
\end{aligned}$$

Since $\sum_{x=1}^{\ell} |P_x|$ is equal to n, $\sum_{x=1}^{\ell} |P_x|^2$ is minimized when each $|P_x|$ is equal to n/ℓ, i.e.

$$\sum_{x=1}^{\ell} |P_x|^2 \geq \sum_{x=1}^{\ell} \left(\frac{n}{\ell}\right)^2 = \frac{n^2}{\ell}.$$

Substituting it above:

$$\mathbb{E}[M_a] \leq \frac{q}{2} \left(n^2 - \frac{n^2}{\ell}\right) = \frac{qn^2}{2} \left(1 - \frac{1}{\ell}\right).$$

Let $p_m = \max_{e \in E} p(e)$. Remember that each cross edge e is activated with probability $p(e)$ ($\leq p_m$). So, we have

$$\mathbb{E}\left[\max_{\boldsymbol{S}} \left(\sum_{x=1}^{\ell} \mathcal{I}(S_{P_x}) - \mathcal{I}(\boldsymbol{S})\right)\right] \leq 2 \cdot \mathbb{E}[M_a] \cdot p_m$$

And therefore,

$$\mathbb{E}\left[\max_{\boldsymbol{S}}\left(\sum_{x=1}^{\ell}\mathcal{I}(S_{P_x})-\mathcal{I}(\boldsymbol{S})\right)\right]\leq qn^2\left(1-\frac{1}{\ell}\right)p_m$$

Also, note that $\sum_{x=1}^{\ell}\mathcal{I}(S_{P_x})$ is always at least as large as $\mathcal{I}(\boldsymbol{S})$, i.e. $\sum_{x=1}^{\ell}\mathcal{I}(S_{P_x})-\mathcal{I}(\boldsymbol{S})\geq 0$. This gives us the desired result:

$$\mathbb{E}\left[\max_{\boldsymbol{S}}\left|\sum_{x=1}^{\ell}\mathcal{I}(S_{P_x})-\mathcal{I}(\boldsymbol{S})\right|\right]\leq qn^2\left(1-\frac{1}{\ell}\right)p_m$$

F.4 Proof of Lemma 7.5.2 & Lemma 7.5.3

We go over the exact procedure of the modified VE algorithm and prove Lemmas 7.5.2 and 7.5.3 in the process. For the forward pass, we compute $\max_{\vec{a}}\sum_{x=1}^{\ell}f_x(\vec{a}_x)+f_c(\vec{a})$. We know that f_c depends only on the L-1 norm of $\vec{a}$, so we represent it as $f_c(\|\vec{a}\|_1)$. Also note that, the communities are disjoint, because of which each action bit a_i (of action $\vec{a}$) appears in the argument of exactly one factor f_x (other than the constraint factor f_c).

As mentioned in the paper, we eliminate all variables of a community at once. So, to eliminate the first block of variables, we compute $\max_{\vec{a}_1}f_1(\vec{a}_1)+f_c(\|\vec{a}\|_1)=\psi_1(\|\vec{a}_{-1}\|_1)$, where $\vec{a}_{-1}$ denotes all action bits of $\vec{a}$ except those in $\vec{a}_1$. Note that, in the RHS of this expression, we use $\|\vec{a}_{-1}\|_1$ as opposed to $\vec{a}_{-1}$ itself because the LHS (before computing the max) depends only on $\vec{a}_1$ and $\|\vec{a}_1\|_1+\|\vec{a}_{-1}\|_1$. Also, note that for $\|\vec{a}_{-1}\|_1>z$, we have $\|\vec{a}\|_1>z$ making $f_c(\|\vec{a}\|_1)$ and $\psi_1(\|\vec{a}_{-1}\|_1)$ equal to $-\infty$.

To make this more concrete, Table F.1 shows how ψ_1 is exactly computed. Here, $v_i^{(x)}$ denotes the maximum value of f_x when exactly i bits of $\vec{a}_x$ are 1, and s_x denotes the number of bits in $\vec{a}_x$.

$\|\vec{a}_{-1}\|_1$	$\psi_1(\|\vec{a}_{-1}\|_1)$
0	$\max\left(v_0^{(1)}+f_c(0), v_1^{(1)}+f_c(1), \cdots v_{s_1}^{(1)}+f_c(s_1)\right)$
1	$\max\left(v_0^{(1)}+f_c(1), v_1^{(1)}+f_c(2), \cdots v_{s_1}^{(1)}+f_c(s_1+1)\right)$
$\vdots$	$\vdots$
z	$v_0^{(1)}+f_c(z)$
$>z$	$-\infty$

Table F.1: Factor obtained on (first) block elimination

Apart from computing the maximum objective value (forward pass), we also need to compute the maximizing assignment of the problem (backward pass). For this, we maintain another function $\mu_1(\|\vec{a}_{-1}\|_1)$ which keeps track of the value of $\vec{a}_1$ at which this maximum

is attained (for each value of $\|\vec{a}_{-1}\|_1$), i.e. $\mu_1(v) = argmax_{\vec{a}_1}\left[f_1(\vec{a}_1) + f_c(\|\vec{a}_1\|_1 + v)\right]$. After eliminating variables of the first community, we are left with $\max_{\vec{a}_{-1}} \sum_{x=2}^{\ell} f_x(\vec{a}_x) + \psi_1(\|\vec{a}_{-1}\|_1)$. We repeat the same procedure and eliminate $\vec{a}_2$ by computing $\max_{\vec{a}_2} f_2(\vec{a}_2) + \psi_1(\|\vec{a}_{-1}\|_1)$, to obtain $\psi_2(\|\vec{a}_{-1,-2}\|_1)$. Note that, again, ψ_2 depends only on the L-1 norm of the remaining variables. Also, for $\|\vec{a}_{-1,-2}\|_1 > z$, ψ_2 becomes $-\infty$. In a similar way, this holds for the remaining generated factors, giving Lemma 7.5.2.

Once we complete the forward pass, we are left with $\psi_\ell(0)$ which is the maximum value of the objective function. Then, as in standard VE, we backtrack and use the μ_x functions to obtain the maximizer $argmax_{\vec{a}} \sum_{x=1}^{\ell} f_x(\vec{a}_x) + f_c(\|\vec{a}\|_1)$, i.e. $\mu_\ell(0)$ gives us the value of $\vec{a}_\ell$, then $\mu_{\ell-1}(\|\vec{a}_\ell\|_1)$ gives us the value of $\vec{a}_{\ell-1}$, $\mu_{\ell-2}(\|\vec{a}_\ell\|_1 + \|\vec{a}_{\ell-1}\|_1)$ gives us the value of $\vec{a}_{\ell-2}$ and so on.

Observe that to compute the i^{th} derived factor, we needed to compute $\max_{\vec{a}_i} f_i(\vec{a}_i) + \psi_{i-1}(\|\vec{a}_{-1,-2,\cdots-(i-1)}\|_1) = \psi_i(\|\vec{a}_{-1,-2,\cdots-i}\|_1)$. And for this, we just need to compute $v_s^{(i)}$ for each $s = 0, 1, \cdots s_i$, as evident from Table F.1. This takes time $O(2^{s_i})$, where s_i denotes the size of the i^{th} community. Hence, the time complexity of the whole algorithm is $\sum_{i=1}^{\ell} O\left(2^{s_i}\right)$.

Appendix G

Omitted Proofs and Results for Chapter 9

G.1 IRL Algorithms

In this appendix we identify two well-known algorithms that match feature expectations.

G.1.1 Apprenticeship Learning

Under the classic Apprenticeship Learning algorithm, designed by Abbeel and Ng. [AN04a], a policy $\pi^{(0)}$ is selected to begin with. Its feature expectation $\mu(\pi^{(0)})$ is computed and added to the bag of feature expectations. At each step,

$$t^{(i)} = \max_{\mathbf{w}:\|\mathbf{w}\|_2 \leq 1} \min_{j \in \{0,..,i-1\}} \mathbf{w}^\intercal \left(\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) - \mu\left(\pi^{(j)}\right) \right)$$

is computed along with the weight $\mathbf{w}^{(i)}$ that achieved this. When $t^{(i)} \leq \epsilon$ the algorithm terminates, otherwise the associated optimal policy $\pi^{(i)}$ is computed, and its corresponding feature expectation vector $\mu(\pi^{(i)})$ is added to the bag of feature expectations. The algorithm provides the following guarantee.

Theorem G.1.1 (adapted from [AN04a])**.** *For any $\epsilon > 0$, the Apprenticeship Learning algorithm terminates with $t^{(i)} \leq \epsilon$ after a number of iterations bounded by*

$$T = O\left(\frac{d}{(1-\gamma)^2 \epsilon^2} \ln \frac{d}{(1-\gamma)\epsilon} \right),$$

and outputs a mixture over $\pi^{(1)}, ..., \pi^{(T)}$ that ϵ-matches the feature expectations of the observed trajectories.

Note that it is necessary for us to use a randomized policy, in contrast to the case where a single deterministic policy generated all the trajectory samples, as, in our case, typically there is no single deterministic policy that matches the feature expectations of the observed trajectories.

G.1.2 Max Entropy

We next discuss the Max Entropy algorithm of Ziebart et al. [Zie+08], which optimizes the max entropy of the probability distribution over trajectories subject to the distribution satisfying approximate feature matching. This is done to resolve the potential ambiguity of there being multiple stochastic policies that satisfy feature matching. Optimizing entropy is equivalent to maximizing the regularized likelihood $L(\mathbf{w})$ of the observed trajectories. Specifically, the objective is

$$L(\mathbf{w}) = \max_{\mathbf{w}} \sum_{i=1}^{n} \log \Pr[\tau_i|\mathbf{w}, T] - \sum_{i=1}^{d} \rho_i \|\mathbf{w}_i\|_1,$$

with

$$\Pr[\tau_i|\mathbf{w}, T] = \frac{e^{\mathbf{w}^\intercal \phi(\tau_i)}}{Z(\mathbf{w}, T)} \prod_{s_t, a_t, s_{t+1} \in \tau_i} T(s_t, a_t, s_{t+1}).$$

The regularization term is introduced to allow for approximate feature matching since the observed empirical feature expectation may differ from the true expectation. Let $\boldsymbol{\rho}$ be an upper bound on this difference, i.e., for all $k = 1, \ldots, d$,

$$\rho_k \geq \left| \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i)_k - \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i)_k \right] \right|.$$

One may then derive that the gradient of $L(\mathbf{w})$ is the difference between the feature expectation induced $\mathbf{w}$ and the observed feature expectation.

Theorem G.1.2 (adapted from [Zie+08])**.** *Let $\epsilon > 0$, and assume that the Max Entropy algorithm finds $\mathbf{w}$ such that $|\nabla L(\mathbf{w})| < \epsilon$, then this $\mathbf{w}$ corresponds to a randomized policy that $(\epsilon + \|\boldsymbol{\rho}\|_1)$-matches the feature expectations of the observed trajectories.*

The assumption on the gradient is needed because the above optimization objective is derived only with the approximate feature matching constraint. MDP dynamics is not explicitly encoded into the optimization. Instead, heuristically, the likelihood of each trajectory $\Pr[\tau_i|\mathbf{w}, T]$ is weighted by the product of the transition probabilities of its steps. The follow-up work of Ziebart [Zie10] addresses this by explicitly introducing MDP constraints into the optimization, and optimizing for the causal entropy, thereby achieving unconditional feature matching.

G.2 Proof of Theorem 9.3.2

We need to bound the difference between $R^{\mathbf{w}^\star}(\tilde{\pi})$ and $R^{\mathbf{w}^\star}(\pi^u)$. First, recall that $\tilde{\pi}$ $\epsilon/3-$matches the feature expectations of $\tau_1, \ldots, \tau_n$. It holds that

$$\begin{aligned} \left| R^{\mathbf{w}^\star}(\tilde{\pi}) - (\mathbf{w}^\star)^\intercal \left(\frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right) \right| &= \left| (\mathbf{w}^\star)^\intercal \left(\mu(\tilde{\pi}) - \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right) \right| \\ &\leq \|\mathbf{w}^\star\|_2 \left\| \mu(\tilde{\pi}) - \frac{1}{n} \sum_{i=1}^{n} \phi(\tau_i) \right\|_2 \leq \frac{\epsilon}{3}, \end{aligned} \tag{G.1}$$

where the second transition follows from the Cauchy-Schwarz inequality, and the last from the assumption that $\|\mathbf{w}^\star\|_2 \leq 1$. Hence, it is sufficient to demonstrate that, with probability at least $1-\delta$,

$$\left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - R^{\mathbf{w}^\star}(\pi^u)\right| \leq \frac{2\epsilon}{3}, \tag{G.2}$$

as the theorem would then follow from Equations (G.1), and (G.2) by the triangle inequality.

We note that the difference on the left hand side of Equation (G.2) is due to two sources of noise.

1. The finite number of samples of trajectories which, in our setting, originates from multiple policies.
2. The truncated trajectories τ_i which are limited to L steps.

Formally, let τ_i' denote the infinite trajectory for each i, then the difference can be written as

$$\begin{aligned}\left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - R^{\mathbf{w}^\star}(\pi^u)\right| \leq{}& \left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right) - (\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right| \\ &+ \left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right) - R^{\mathbf{w}^\star}(\pi^u)\right|\end{aligned}$$

Bounding finite sample noise. We wish to bound:

$$\left|(\mathbf{w}^\star)^\intercal \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right) - R^{\mathbf{w}^\star}(\pi^u)\right| = \left|\frac{1}{n}\left(\sum_{i=1}^{n}(\mathbf{w}^\star)^\intercal(\phi(\tau_i') - \mu(\pi_i))\right)\right|. \tag{G.3}$$

Define random variable $Z_i = (\mathbf{w}^\star)^\intercal(\phi(\tau_i') - \mu(\pi_i))$. Then the right-hand side of Equation (G.3) may be expressed as $|\frac{1}{n}\sum_{i=1}^{n} Z_i|$. Furthermore, Z_i is such that $\mathbb{E}[\phi(\tau_i')_k] = \mu(\pi_i)_k$ for all $k = 1, \ldots, d$. This is because a policy π_i defines a distribution over trajectories, and τ_i' is a draw from this distribution. Using the linearity of expectation, it follows that

$$\mathbb{E}[Z_i] = (\mathbf{w}^\star)^\intercal \mathbb{E}[\phi(\tau_i') - \mu(\pi_i)] = 0.$$

Moreover,

$$|Z_i| \leq \|\mathbf{w}^\star\|_2 \|\phi(\tau_i')\|_2 + \|\mathbf{w}^\star\|_2 \|\mu(\pi_i)\|_2 \leq \frac{2\sqrt{d}}{1-\gamma},$$

since $\|\phi(s,\cdot)\|_\infty = 1$. Thus, using Hoeffding's inequality, we conclude that

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} Z_i\right| > \frac{\epsilon}{3}\right] \leq 2\exp\left(-\frac{2n\left(\frac{\epsilon}{3}\right)^2}{(\frac{4\sqrt{d}}{1-\gamma})^2}\right) \leq \delta,$$

where the last transition holds by our choice of n.

Bounding bias due to truncated trajectories. We wish to bound:

$$\left|(\mathbf{w}^\star)^\intercal\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right)-(\mathbf{w}^\star)^\intercal\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right|.$$

For each trajectory τ_i, truncating after L steps incurs a reward difference of:

$$\begin{aligned}|(\mathbf{w}^\star)^\intercal\phi(\tau_i')-(\mathbf{w}^\star)^\intercal\phi(\tau_i)| &= \left|(\mathbf{w}^\star)^\intercal\sum_{t=L}^{\infty}\gamma^t\phi(\tau_i'(s_t),\tau_i'(a_t))\right| \\ &\leq \sum_{t=L}^{\infty}\gamma^t\|\mathbf{w}^\star\|_2\|\phi(\tau_i'(s_t),\tau_i'(a_t))\|_2 \leq \gamma^L\frac{\sqrt{d}}{1-\gamma}\leq\frac{\epsilon}{3},\end{aligned}$$

where the third transition holds because $\|\phi(\tau_i(s_t),\tau_i(a_t))\|_2\leq\sqrt{d}$, and the last transition follows from our choice of L. Hence, we obtain

$$\left|(\mathbf{w}^\star)^\intercal\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i)\right)-(\mathbf{w}^\star)^\intercal\left(\frac{1}{n}\sum_{i=1}^{n}\phi(\tau_i')\right)\right|\leq\frac{1}{n}\sum_{i=1}^{n}|(\mathbf{w}^\star)^\intercal\phi(\tau_i)-(\mathbf{w}^\star)^\intercal\phi(\tau_i')|\leq\frac{\epsilon}{3}.$$

□

G.3 Proof of Theorem 9.4.1

We require a key property of sub-exponential random variables, which is captured by the following well known tail inequality; its proof can be found, for example, in Chapter 2 of [Wai19].

Lemma G.3.1. *Let $X_1,\ldots,X_m$ be independent sub-exponential random variables with parameters (ν,b). Then*

$$\Pr\left[\frac{1}{m}\sum_{j=1}^{m}(X_j-u_j)\geq t\right]\leq\begin{cases}\exp\left(-\frac{mt^2}{2\nu^2}\right) & \text{for } 0\leq t\leq\frac{\nu^2}{b}\\ \exp\left(-\frac{mt}{2b}\right) & \text{for } t>\frac{\nu^2}{b}\end{cases},$$

where $u_j=\mathbb{E}[X_j]$.

Turning to the theorem's proof, as π^u is a uniform distribution over the policies $\pi_1,\ldots,\pi_n$, its expected reward is given by

$$R^{\mathbf{w}^\star}(\pi^u)=\frac{1}{n}\sum_{i=1}^{n}R^{\mathbf{w}^\star}(\pi_i). \tag{G.4}$$

Observe that $R^{\mathbf{w}^\star}(\pi_i)$ is a random variable which is i.i.d. across i, as the corresponding noise $\boldsymbol{\eta}_i$ is i.i.d. as well. We analyze the expectation of the difference with respect to $R^{\mathbf{w}^\star}(\pi^\star)$.

First, note that for a weight vector $\mathbf{w}$ and policy π,

$$R^{\mathbf{w}}(\pi)=\sum_{t=0}^{\infty}\gamma^t\sum_{s\in S}P_\pi(s,t)\mathbf{w}^\intercal\phi(s,\pi(s)), \tag{G.5}$$

where $P_\pi(s,t)$ denotes the probability of being in state s on executing policy π from the start. Hence, for each $i \in N$, we have

$$
\begin{aligned}
&R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i) \\
&\quad= \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[P_{\pi^\star}(s,t)(\mathbf{w}^\star)^\intercal \phi(s, \pi^\star(s)) - P_{\pi_i}(s,t)(\mathbf{w}^\star)^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad= \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[P_{\pi^\star}(s,t)(\mathbf{w}_i - \boldsymbol{\eta}_i)^\intercal \phi(s, \pi^\star(s)) - P_{\pi_i}(s,t)(\mathbf{w}_i - \boldsymbol{\eta}_i)^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad= R^{\mathbf{w}_i}(\pi^\star) - R^{\mathbf{w}_i}(\pi_i) + \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi^\star(s)) + P_{\pi_i}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad\leq \sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi^\star(s)) + P_{\pi_i}(s,t) \boldsymbol{\eta}_i^\intercal \phi(s, \pi_i(s)) \Big] \\
&\quad= \sum_{k=1}^{d} \eta_{ik} \left[\sum_{t=0}^{\infty} \gamma^t \sum_{s \in S} \Big[- P_{\pi^\star}(s,t) \phi(s, \pi^\star(s))_k + P_{\pi_i}(s,t) \phi(s, \pi_i(s))_k \Big] \right] \\
&\quad\coloneqq \sum_{k=1}^{d} \eta_{ik} \alpha_{ik}, \qquad \text{(G.6)}
\end{aligned}
$$

where the inequality holds since $R^{\mathbf{w}_i}(\pi_i) \geq R^{\mathbf{w}_i}(\pi^\star)$, which, in turn, holds because π_i is optimal under $\mathbf{w}_i$.

Using the assumption that $\|\phi(s,a)\|_\infty \leq 1$, it holds that $\left|\sum_{s \in S} P_\pi(s,t) \phi(s,a)_k\right| \leq 1$ for any policy π. We can therefore bound $|\alpha_{ik}|$ as follows.

$$
\begin{aligned}
|\alpha_{ik}| &\leq \sum_{t=0}^{\infty} \gamma^t \left| \sum_{s \in S} [-P_{\pi^\star}(s,t) \phi(s, \pi^\star(s))_k + P_{\pi_i}(s,t) \phi(s, \pi_i(s))_k] \right| \\
&\leq \sum_{t=0}^{\infty} \gamma^t \left[\left| \sum_{s \in S} P_{\pi^\star}(s,t) \phi(s, \pi^\star(s))_k \right| + \left| \sum_{s \in S} P_{\pi_i}(s,t) \phi(s, \pi_i(s))_k \right| \right] \\
&\leq \frac{2}{1-\gamma}.
\end{aligned}
$$

Therefore, it holds that

$$
\|\boldsymbol{\alpha}_i\|_2 = \sqrt{\sum_{k=1}^{d} \alpha_{ik}^2} \leq \sqrt{\sum_{k=1}^{d} \left(\frac{2}{1-\gamma} \right)^2} = \frac{2\sqrt{d}}{(1-\gamma)}.
$$

Using this bound along with Equation (G.6), we obtain

$$
R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i) \leq \sum_{k=1}^{d} \eta_{ik} \alpha_{ik} \leq \|\boldsymbol{\eta}_i\|_2 \|\boldsymbol{\alpha}_i\|_2 \leq \frac{2\sqrt{d}}{(1-\gamma)} \sqrt{\sum_{k=1}^{d} \eta_{ik}^2}
$$

$$= \frac{2d}{(1-\gamma)} \sqrt{\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2}. \tag{G.7}$$

Denote $u = \mathbb{E}[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2]$. To compute the expected value of the previous expression (with respect to the randomness of the noise $\boldsymbol{\eta}_i$), we analyze

$$\begin{aligned}
\mathbb{E}\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2}\right] &= \int_0^\infty \Pr\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2} \geq x\right] dx = \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq x^2\right] dx \\
&= \int_0^{\sqrt{u}} \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq x^2\right] dx + \int_{\sqrt{u}}^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq x^2\right] dx \\
&\leq \int_0^{\sqrt{u}} 1 \; dx + \int_{\sqrt{u}}^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq x^2\right] dx \\
&= \sqrt{u} + \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq u + t\right] \frac{1}{2\sqrt{u+t}} dt \\
&\leq \sqrt{u} + \frac{1}{2\sqrt{u}} \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq u + t\right] dt,
\end{aligned}$$

where the fourth transition is obtained by changing the variable using $x = \sqrt{u+t}$. But since each η_{ik}^2 is sub-exponential with parameters (ν, b), from Lemma G.3.1 we have

$$\Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq u + t\right] \leq \begin{cases} \exp\left(-\frac{dt^2}{2\nu^2}\right) & \text{for } 0 \leq t \leq \frac{\nu^2}{b} \\ \exp\left(-\frac{dt}{2b}\right) & \text{for } t > \frac{\nu^2}{b} \end{cases}.$$

Plugging this into the upper bound for the expected value gives us

$$\begin{aligned}
\mathbb{E}\left[\sqrt{\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2}\right] &\leq \sqrt{u} + \frac{1}{2\sqrt{u}} \int_0^\infty \Pr\left[\frac{1}{d}\sum_{k=1}^{d} \eta_{ik}^2 \geq u + t\right] dt \\
&\leq \sqrt{u} + \frac{1}{2\sqrt{u}} \left[\int_0^{\frac{\nu^2}{b}} \exp\left(-\frac{dt^2}{2\nu^2}\right) dt + \int_{\frac{\nu^2}{b}}^\infty \exp\left(-\frac{dt}{2b}\right) dt\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}} \left[\int_0^{\frac{\nu\sqrt{d}}{b}} \exp\left(-\frac{z^2}{2}\right) \frac{\nu}{\sqrt{d}} dz + \left(-\frac{2b}{d}\right) \exp\left(-\frac{dt}{2b}\right) \Big|_{\frac{\nu^2}{b}}^\infty\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}} \left[\sqrt{\frac{2\pi}{d}}\nu \int_0^{\frac{\nu\sqrt{d}}{b}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz + \frac{2b}{d} \exp\left(-\frac{d\nu^2}{2b^2}\right)\right] \\
&= \sqrt{u} + \frac{1}{2\sqrt{u}} \left[\sqrt{\frac{2\pi}{d}}\nu \left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{2b}{d} \exp\left(-\frac{d\nu^2}{2b^2}\right)\right]
\end{aligned}$$

$$= \sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu\left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right), \tag{G.8}$$

where the transition in the third line is obtained by changing the variable using $t = \frac{v}{\sqrt{d}}z$, and Φ denotes the CDF of a standard normal distribution. Hence, taking an expected value for Equation (G.7) and plugging in Equation (G.8), we obtain

$$\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi^\star) - R^{\mathbf{w}^\star}(\pi_i)\right] \leq \frac{2d}{(1-\gamma)}\left[\sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu\left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right].$$

Rearranging this equation, we have

$$\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi_i)\right] \geq R^{\mathbf{w}^\star}(\pi^\star) - \frac{2d}{(1-\gamma)}\left[\sqrt{u} + \sqrt{\frac{\pi}{2ud}}\nu\left(\Phi\left(\frac{\nu\sqrt{d}}{b}\right) - \frac{1}{2}\right) + \frac{b}{d\sqrt{u}}\exp\left(-\frac{d\nu^2}{2b^2}\right)\right].$$

Taking an expectation over Equation (G.4) gives us $\mathbb{E}\left[R^{\mathbf{w}^\star}(\pi^u)\right] = \mathbb{E}\left[R^{\mathbf{w}^\star}(\pi_i)\right]$, and the theorem directly follows. □

We remark that Theorem 9.4.1 can easily be strengthened to obtain a high probability result (at the cost of complicating its statement). Indeed, the reward of the uniform mixture $R^{\mathbf{w}^\star}(\pi^u)$ is the average of the individual policy rewards $R^{\mathbf{w}^\star}(\pi_i)$, which are i.i.d. Further, each of these rewards is bounded, because of the constraints on $\mathbf{w}^\star$ and ϕ. Hence, Hoeffding's inequality would show that $R^{\mathbf{w}^\star}(\pi^u)$ strongly concentrates around its mean.

G.4 Example for the Tightness of Theorem 9.4.1

Assume $\eta_{ik} \sim \mathcal{N}(0, \sigma^2)$ with $\sigma \leq 2/d$ (to avoid violating the constraint $\|\phi(s,a)\|_\infty \leq 1$). Suppose the MDP has just one state and $2^{d-1}+1$ actions. One action has feature vector $(d\sigma/2, 0, \ldots, 0)$, and for each subset $S \subseteq \{2, \ldots, d\}$, there is an action a_S with a binary feature vector such that it is 1 for coordinates in S and 0 everywhere else. Let $w^\star = (1, 0, ..., 0)$. The optimal policy is to pick the first action which has cumulative reward of $\frac{d\sigma}{2(1-\gamma)}$. As $\eta_{ik} \sim \mathcal{N}(0, \sigma^2)$ for each k, with constant probability, roughly $d/2$ of the coordinates of the noised vector reward $\mathbf{w}_i$ will deviate by roughly $+\sigma$ and the first coordinate will not increase too much. In this case, the action corresponding to the coordinates with positive deviations will have reward on the order of $d\sigma/2$, beating action 1 to become optimal. Hence, this would lead to π_i picking this action and having 0 reward under $\mathbf{w}^\star$. As this occurs with constant probability for a policy in the data, and π^u is simply a mean of their rewards, its expected value would deviate from the optimum by at least a constant fraction of $d\sigma/2$.

G.5 Empirical Results for the MDP setting

As we have seen in Section 9.4.1, the gap between $R^{\mathbf{w}^\star}(\pi^\star)$ and $R^{\mathbf{w}^\star}(\pi^u)$ is upper bounded by $O(d\sqrt{u} + \nu\sqrt{d/u} + b/\sqrt{u})$ when η_{ik}^2 is sub-exponential, or $O(d\sigma)$ when η_{ik} is Gaussian.

Further, Section 9.3 shows that a policy $\tilde{\pi}$ that matches feature expectations of the observed trajectories is very close to π^u in terms of cumulative reward $R^{\mathbf{w}^\star}$. In this appendix, we empirically examine the gaps between $\tilde{\pi}$ (obtained by a "feature matching" IRL algorithm), π^u and $\pi^\star$.

G.5.1 Methodology

As our IRL algorithm we use Apprenticeship Learning, which guarantees the feature-matching property (see Section 9.3 and Appendix G.1). By Theorem 9.3.2 we may safely assume that any IRL algorithm that matches feature expectations would have essentially identical rewards, and therefore would show very similar behavior in our experiments.

We perform our experiments in the following two domains.

Grab a Milk. We adapt the "Grab a Milk" MDP, a route planning RL domain [WL18], to our setting. The MDP is defined by a 10 by 10 grid room, where the agent starts at $(0, 0)$ and has to reach a bottle of milk positioned at $(9, 9)$. There are also 16 babies in the room, 5 of which are crying for attention. When the agent crosses a crying baby, they can help soothe the baby, but on crossing a non-crying baby, the agent disturbs the baby. Hence, the goal of this task is to minimize the number of steps to the milk, while at the same time soothing as many crying babies as possible along the way and avoiding crossing non-crying babies. This MDP is adapted to our setting, by defining each state (or grid square) to have three features $\phi(s)$.[1] The first feature captures the reward of taking a step, and is set to -1 if the state is non-terminal, whereas it is set to 5 for the terminal state $(9, 9)$. The second is a boolean feature depicting whether there is a crying baby in the particular grid square, and similarly the third is a boolean feature depicting whether there is a non-crying baby in the particular grid square. The rewards in the MDP are then defined as $r^{\mathbf{w}^\star}(s) = (\mathbf{w}^\star)^\intercal \phi(s)$ where the ground truth weight vector is given by $\mathbf{w}^\star = [1, 0.5, -0.5]$. Intuitively, this weight vector $\mathbf{w}^\star$ can be interpreted as the weights for different ethical factors, and each member of society has a noised version of this weight.

Sailing. The other domain we use is a modified version of the "Sailing" MDP [KS06]. The Sailing MDP is also a gridworld domain (we use the same size of 10 by 10), where there is a sailboat starting at $(0, 0)$ and navigating the grid under fluctuating wind conditions. The goal of the MDP is to reach a specified grid square as quickly as possible. We adapt this domain to our setting by removing the terminal state, and instead adding features for each grid square.[2] Now, the goal of the agent is not to reach a certain point as quickly as possible, but to navigate this grid while maximizing (or minimizing) the weighted sum of these features. We use 10 features for each grid square, and these are independently sampled from a uniform distribution over $(-1, 1)$. The ground truth weight vector $\mathbf{w}^\star$, which defines the weights of these features for the net reward, is also randomly sampled from independent Unif$(-1, 1)$ for each coordinate. As before, this weight vector $\mathbf{w}^\star$ can be

[1] For these MDPs, the rewards depend only on the states and not state-action pairs, and hence the reward function can be defined as $r^{\mathbf{w}}(s, a) = r^{\mathbf{w}}(s) = \mathbf{w}^\intercal \phi(s)$.

[2] Intuitively, these features could represent aspects like "abundance of fish" in that grid square for fishing, "amount of trash" in that square that could be cleaned up, "possible treasure" for treasure hunting, etc.

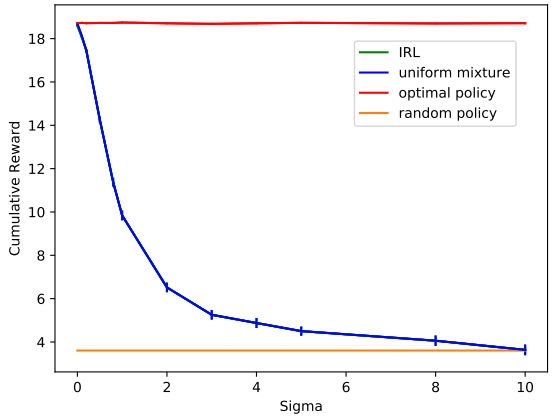


Figure G.1: Performance on the Sailing MDP. Error bars show 95% confidence intervals.

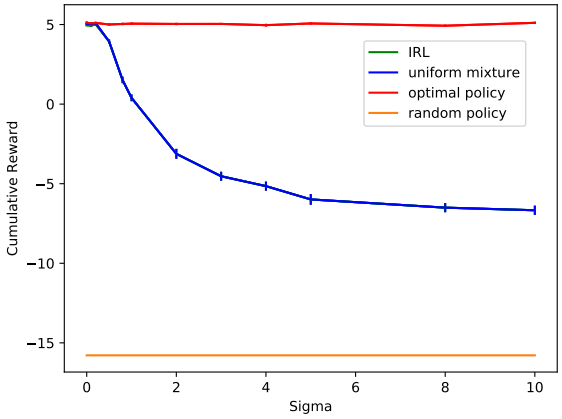


Figure G.2: Performance on the Grab a Milk MDP. Error bars show 95% confidence intervals.

interpreted as the weights for different bounties, and each member has a noised version of this weight.

Being gridworld domains, in both the MDPs, the agent has four actions to choose from at each state (one for each direction). The transition dynamics are as follows: On taking a particular action from a given state, the agent moves in that direction with probability 0.95, but with a probability of 0.05 it moves in a different direction uniformly at random. We use a discount factor of 0.95 in both domains.

We generate the trajectories $\{\tau_1, \ldots, \tau_n\}$ as described in Section 9.3, and use a Gaussian distribution for the noise. That is, $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \sigma^2 I_d)$. We generate a total of $n = 50$ trajectories, each of length $L = 30$. IRL is then performed on this data and we analyze its reward as σ is varied. A learning rate of 0.001 is used for the Apprenticeship Learning algorithm.

G.5.2 Results

Figures G.1 and G.2 show the performance of π^u and the IRL algorithm as σ is varied. We also include the performance of $\pi^\star$ and a purely random policy π^r (which picks a uniformly random action at each step), as references. Each point in these graphs is averaged over 50 runs (of data generation).

For both domains, the first thing to note is that the uniform mixture π^u and the IRL algorithm have nearly identical rewards, which is why the green IRL curve is almost invisible. This confirms that matching feature expectations leads to performance approximating the uniform mixture.

Next, as expected, one can observe that as σ increases, the gap between $R^\star(\pi^\star)$ and $R^\star(\pi^u)$ also increases. Further, for both domains, this gap saturates around $\sigma = 10$ and the $R^\star(\pi^u)$ curve flattens from there (hence, we do not include larger values of σ in either graph). Note that, in both domains, the ground truth weight vector $\mathbf{w}^\star$ is generated such

that $\|\mathbf{w}^\star\|_\infty \leq 1$. Hence, a standard deviation of 10 in the noise overshadows the true weight vector $\mathbf{w}^\star$, leading to the large gap shown in both graphs. Looking at more reasonable levels of noise (with respect to the norm of the weights), like $\sigma \in [0, 1]$, we can see that $R^\star(\pi^u)$ drops approximately linearly, as suggested by Theorem 9.4.1. In particular, it is 14.27 at $\sigma = 0.5$ and 9.84 at $\sigma = 1.0$ for Sailing, and it is 3.93 at $\sigma = 0.5$ and 0.39 at $\sigma = 1.0$ for Grab a Milk.

Finally, we compare the performance of π^u with that of the purely random policy π^r. As σ becomes very large, each $\mathbf{w}_i$ is distributed almost identically across the coordinates. Nevertheless, because of the structure of the Grab a Milk MDP, $R^\star(\pi^u)$ still does significantly better than $R^\star(\pi^r)$. By contrast, Sailing has features that are sampled i.i.d. from Unif$(-1, 1)$ for each state, which leads the two policies, π^u and π^r, to perform similarly for large values of σ.

G.6 Proof of Lemma 9.4.2

Before proving the lemma, we look at a relatively simple example that we will use later to complete the proof.

G.6.1 Simpler Example

Consider an MDP with a single state s, and three actions $\{a, b, c\}$. Since s is the only state, $T(s, a, s) = T(s, b, s) = T(s, c, s) = 1$, and D is degenerate at s. This implies that there are only three possible policies, denoted by π_a, π_b, π_c (which take actions a, b, c respectively from s). Let the feature expectations be

$$\begin{aligned}\phi(s, a) &= [0.5, 0.5],\\ \phi(s, b) &= [1, -\delta/2],\\ \phi(s, c) &= [-\delta/2, 1],\end{aligned}$$

where $\delta > 0$ is a parameter. Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\begin{aligned}\mu_a &= \frac{1}{2(1-\gamma)}[1, 1],\\ \mu_b &= \frac{1}{2(1-\gamma)}[2, -\delta],\\ \mu_c &= \frac{1}{2(1-\gamma)}[-\delta, 2].\end{aligned}$$

Let the ground truth weight vector be $\mathbf{w}^\star = (v_o, v_o)$, where v_o is a "large enough" positive constant. In particular, v_o is such that the noised weight vector $\mathbf{w} = \mathbf{w}^\star + \boldsymbol{\eta}$ has probability strictly more than $1/3$ of lying in the first quadrant. For concreteness, set v_o to be such that $\Pr(\mathbf{w} > 0) = 1/2$. Such a point always exists for any noise distribution (that is continuous and i.i.d. across coordinates). Specifically, it is attained at $v_o = -F^{-1}(1 - \frac{1}{\sqrt{2}})$,

where F^{-1} is the inverse CDF of each coordinate of the noise distribution. This is because at this value of v_o,

$$\begin{aligned}\Pr(\mathbf{w} > 0) &= \Pr((v_o, v_o) + (\eta_1, \eta_2) > 0) = \Pr(v_o + \eta_1 > 0)^2 \\ &= \Pr(\eta_1 > -v_o)^2 = (1 - F(-v_o))^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2}.\end{aligned}$$

Let us look at weight vectors $\mathbf{w}$ for which each of the three policies π_a, π_b and π_c are optimal. π_a is the optimal policy when $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_b$ and $\mathbf{w}^\intercal \mu_a > \mathbf{w}^\intercal \mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) > 0$ and $\mathbf{w}^\intercal(1+\delta, -1) > 0$. On the other hand, π_b is optimal when $\mathbf{w}^\intercal \mu_b > \mathbf{w}^\intercal \mu_a$ and $\mathbf{w}^\intercal \mu_b > \mathbf{w}^\intercal \mu_c$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(-1, 1+\delta) < 0$ and $\mathbf{w}^\intercal(1, -1) > 0$. Finally, π_c is optimal when $\mathbf{w}^\intercal \mu_c > \mathbf{w}^\intercal \mu_a$ and $\mathbf{w}^\intercal \mu_c > \mathbf{w}^\intercal \mu_b$, which is the intersection of the half-spaces $\mathbf{w}^\intercal(1+\delta, -1) < 0$ and $\mathbf{w}^\intercal(1, -1) < 0$. These regions are illustrated in Figure 9.1 for different values of δ. Informally, as δ is decreased, the lines separating (π_a, π_c) and (π_a, π_b) move closer to each other (as shown for $\delta = 0.25$), while as δ is increased, these lines move away from each other (as shown for $\delta = 10$).

Formally, let R_δ denote the region of $\mathbf{w}$ for which π_a is optimal (i.e. the blue region in the figures), that is,

$$R_\delta = \left\{ \mathbf{w} : \frac{w_1}{1+\delta} < w_2 < w_1(1+\delta) \right\}.$$

This is bounded below by the line $w_1 = (1+\delta)w_2$, which makes an angle of $\theta_\delta = \text{Tan}^{-1}(\frac{1}{1+\delta})$ with the x-axis, and bounded above by the line $w_2 = (1+\delta)w_1$, which makes an angle of θ_δ with the y-axis. We first show that for any value of δ, the regions of π_b and π_c have the exact same probability. The probability that π_b is optimal is the probability of the orange region which is

$$\begin{aligned}\Pr(\pi_b \text{ is optimal}) &= \int_{-\infty}^{0} \int_{-\infty}^{w_1} \Pr(\mathbf{w}) dw_2 dw_1 + \int_{0}^{\infty} \int_{-\infty}^{\frac{w_1}{(1+\delta)}} \Pr(\mathbf{w}) dw_2 dw_1 \\ &= \int_{-\infty}^{0} \int_{-\infty}^{t_2} \Pr(t_2, t_1) dt_1 dt_2 + \int_{0}^{\infty} \int_{-\infty}^{\frac{t_2}{(1+\delta)}} \Pr(t_2, t_1) dt_1 dt_2 \\ &= \int_{-\infty}^{0} \int_{-\infty}^{t_2} \Pr(t_1, t_2) dt_1 dt_2 + \int_{0}^{\infty} \int_{-\infty}^{\frac{t_2}{(1+\delta)}} \Pr(t_1, t_2) dt_1 dt_2 \\ &= \Pr(\pi_c \text{ is optimal}),\end{aligned}$$

where the second equality holds by changing the variables as $t_1 = w_2$ and $t_2 = w_1$, and the third one holds because the noise distribution is i.i.d. across the coordinates. Hence, we have

$$\Pr(\pi_b \text{ is optimal}) = \Pr(\pi_c \text{ is optimal}) = \frac{1 - \Pr(R_\delta)}{2},$$

as R_δ denotes the region where π_a is optimal.

Finally, we show that there exists a value of δ such that $\Pr(R_\delta) = 1/3$. Observe that as $\delta \to 0$, the lines bounding the region R_δ make angles that approach $Tan^{-1}(1) = \pi/4$ and the

two lines touch, causing the region to have zero probability. On the other hand, as $\delta \to \infty$, the angles these lines make approach $Tan^{-1}(0) = 0$, so the region coincides with the first quadrant in the limit. Based on our selection of v_o, the probability of this region is exactly $1/2$. Hence, as δ varies from 0 to ∞, the probability of the region R_δ changes from 0 to $1/2$. Next, note that as $\theta_\delta = \text{Tan}^{-1}(\frac{1}{1+\delta})$, this angle changes continuously as δ changes, and hence does the region R_δ. Finally, as the noise distribution is continuous, the probability of this region R_δ also changes continuously as δ is varied. That is, $\lim_{\epsilon \to 0} \Pr(R_{\delta+\epsilon}) = \Pr(R_\delta)$. Coupling this with the fact that $\Pr(R_\delta)$ changes from 0 to $1/2$ as δ changes from 0 to ∞, it follows that there exists a value of δ in between such that $\Pr(R_\delta)$ is exactly $1/3$. Denote this value of δ by δ_o.

We conclude that for $\mathbf{w}^\star = (v_o, v_o)$ and our MDP construction with $\delta = \delta_o$, $\mathcal{P}(\mathbf{w}^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

G.6.2 Completing the Proof

Consider the same MDP as in Section G.6.1. However, for this example, let the feature expectations be

$$\begin{aligned}
\phi(s,a) &= [0.5, 0.5 \quad , \ -\delta_o/2, 1], \\
\phi(s,b) &= [1, -\delta_o/2, \quad 0.5, 0.5], \\
\phi(s,c) &= [-\delta_o/2, 1, \quad 1, -\delta_o/2],
\end{aligned}$$

where δ_o is as defined in Section G.6.1. Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\begin{aligned}
\mu_a &= \frac{1}{2(1-\gamma)}[1, 1 \quad , \ -\delta_o, 2], \\
\mu_b &= \frac{1}{2(1-\gamma)}[2, -\delta_o, \quad 1, 1], \\
\mu_c &= \frac{1}{2(1-\gamma)}[-\delta_o, 2, \quad 2, -\delta_o].
\end{aligned}$$

Consider two weight vectors $\mathbf{w}_a^\star = (v_o, v_o, 0, 0)$ and $\mathbf{w}_b^\star = (0, 0, v_o, v_o)$, where v_o is as defined in Section G.6.1. Since $\mathbf{w}_a^\star$ completely discards the last two coordinates, it immediately follows from the example of Section G.6.1 that $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Similarly, the same analysis on the last two coordinates shows that $\mathcal{P}(\mathbf{w}_b^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ as well. On the other hand, the optimal policy according to $\mathbf{w}_a^\star$ is π_a while the optimal policy according to $\mathbf{w}_b^\star$ is π_b. Hence, $\pi_a^\star \neq \pi_b^\star$, but we still have $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star)$, leading to non-identifiability. □

G.7 Proof of Theorem 9.4.3

The proof of this theorem strongly relies on Lemma 9.4.2 and the example used to prove it. Consider the MDP as in Section G.6.2, but now with 6 features instead of just 4. In

particular, let the feature expectations of the three policies be

$$\begin{aligned}\phi(s,a) &= [0.5, 0.5 \quad , \ -\delta_o/2, 1, \quad 1, -\delta_o/2],\\ \phi(s,b) &= [1, -\delta_o/2, \quad 0.5, 0.5 \quad , \ -\delta_o/2, 1],\\ \phi(s,c) &= [-\delta_o/2, 1, \quad 1, -\delta_o/2, \quad 0.5, 0.5 \ ].\end{aligned}$$

Hence, the feature expectations of the policies $\{\pi_a, \pi_b, \pi_c\}$ are respectively

$$\begin{aligned}\mu_a &= \frac{1}{2(1-\gamma)}[1, 1 \quad , \ -\delta_o, 2, \quad 2, -\delta_o],\\ \mu_b &= \frac{1}{2(1-\gamma)}[2, -\delta_o, \quad 1, 1 \quad , -\delta_o, 2],\\ \mu_c &= \frac{1}{2(1-\gamma)}[-\delta_o, 2, \quad 2, -\delta_o, \ 1, 1 \quad].\end{aligned}$$

Consider three weight vectors

$$\begin{aligned}\mathbf{w}_a^\star &= (v_o, v_o, 0, 0, 0, 0),\\ \mathbf{w}_b^\star &= (0, 0, v_o, v_o, 0, 0),\\ \mathbf{w}_c^\star &= (0, 0, 0, 0, v_o, v_o).\end{aligned}$$

Since $\mathbf{w}_a^\star$ completely discards the last four coordinates, the example of Section G.6.1 shows that $\mathcal{P}(\mathbf{w}_a^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Similarly, the same analysis on the middle two and last two coordinates shows that $\mathcal{P}(\mathbf{w}_b^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $\mathcal{P}(\mathbf{w}_c^\star) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, respectively. However, the optimal policy according to $\mathbf{w}_a^\star$ is π_a, according to $\mathbf{w}_b^\star$ it is π_b, and according to $\mathbf{w}_c^\star$ it is π_c.

Now, consider an arbitrary algorithm $\mathcal{A}$, which takes as input a distribution over policies and outputs a (possibly randomized) policy. Look at the randomized policy $\mathcal{A}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ returned by $\mathcal{A}$ when the input is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, and let p_a, p_b, p_c be the probabilities it assigns to playing π_a, π_b and π_c. Let p_i (where $i \in \{a, b, c\}$) denote the smallest probability among the three. Then, $p_i \leq 1/3$. Pick the ground truth weight vector to be $\mathbf{w}_i^\star$. As $\mathcal{P}(\mathbf{w}_a^\star) = \mathcal{P}(\mathbf{w}_b^\star) = \mathcal{P}(\mathbf{w}_c^\star)$, the data generated by $\mathbf{w}_i^\star$ follows the distribution $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$, and the policy distribution chosen by $\mathcal{A}$ is simply (p_a, p_b, p_c).

Now, with probability $p_i \leq 1/3$, the policy played is π_i leading to a reward of $\mathbf{w}_i^{\star\intercal}\mu_i = \frac{v_o}{(1-\gamma)}$, and with probability $(1 - p_i)$, the policy played is some π_j (where $j \neq i$) leading to a reward of $\mathbf{w}_i^{\star\intercal}\mu_j = \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)}$ (which is independent of the value of j).[3] Hence, the expected reward of algorithm $\mathcal{A}$ in this case is

$$\begin{aligned}p_i \cdot \frac{v_o}{(1-\gamma)} + (1-p_i) \cdot \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)} &= \frac{(2-\delta_o)}{2}\frac{v_o}{(1-\gamma)} + p_i \cdot \frac{\delta_o}{2}\frac{v_o}{(1-\gamma)}\\ &\leq \frac{(2-\delta_o)v_o}{2(1-\gamma)} + \frac{\delta_o v_o}{6(1-\gamma)}.\end{aligned}$$

[3] An interesting point to note is that by carefully selecting v_o, one could get the corresponding δ_o to be arbitrarily large, thereby causing the optimal and suboptimal policies to have a much larger gap (equally affecting the uniform mixture π^u as well).

Observe that the uniform mixture π^u in this case is just the input distribution $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Whatever be the chosen $\mathbf{w}_i^\star$, the expected reward of this distribution is exactly

$$\frac{1}{3} \cdot \frac{v_o}{(1-\gamma)} + \frac{2}{3} \cdot \frac{(2-\delta_o)}{2} \frac{v_o}{(1-\gamma)} = \frac{(2-\delta_o)v_o}{2(1-\gamma)} + \frac{\delta_o v_o}{6(1-\gamma)},$$

which is nothing but the upper bound on the expected reward of $\mathcal{A}$. Hence, for any algorithm $\mathcal{A}$ there exists a ground truth weight vector $\mathbf{w}_i^\star$ such that $\mathcal{A}$ has an expected reward at most that of π^u (which in turn is strictly suboptimal). □

G.8 Proof of Theorem 9.5.1

To see that this problem is convex, let's analyze the distribution $\mathcal{Q}(\mathbf{w})$.

$$\begin{aligned}
\mathcal{Q}(\mathbf{w})_k &= \Pr(\text{Arm } k \text{ is optimal under weight } (\mathbf{w}+\boldsymbol{\eta})) \\
&= \Pr((\mathbf{w}+\boldsymbol{\eta})^\intercal \mathbf{x}_k \geq (\mathbf{w}+\boldsymbol{\eta})^\intercal \mathbf{x}_j \text{ for all j}) \\
&= \Pr((\mathbf{w}+\boldsymbol{\eta})^\intercal (\mathbf{x}_k - \mathbf{x}_j) \geq 0 \text{ for all j}) \\
&= \Pr(X_k(\mathbf{w}+\boldsymbol{\eta}) \geq 0) \\
&= \Pr(-X_k\boldsymbol{\eta} \leq X_k\mathbf{w}). \qquad \text{(G.9)}
\end{aligned}$$

Since $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I_d)$, we have

$$-X_k\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 X_k X_k^\intercal).$$

And since $X_k X_k^\intercal$ is invertible, this distribution is non-degenerate and has a PDF. Let us use F_k to denote its CDF. Equation (G.9) then reduces to $\mathcal{Q}(\mathbf{w})_k = F_k(X_k\mathbf{w})$. Plugging this back into our optimization problem (9.1), we have

$$\min_{\mathbf{w}} -\sum_{k \in A} \tilde{\mathcal{Q}}_k \log F_k(X_k\mathbf{w}). \qquad \text{(G.10)}$$

As F_k corresponds to a (multivariate) Gaussian which has a log-concave PDF, this CDF is also log-concave. Hence, $\log F_k(X_k\mathbf{w})$ is concave in $\mathbf{w}$ for each k, and therefore (G.10) is a convex optimization problem. □

G.9 Gradient Calculation

From Equation (G.10), we know that the objective function of problem (9.1) can be rewritten as $f(\mathbf{w}) = -\sum_{k \in A} \tilde{\mathcal{Q}}_k \log F_k(X_k\mathbf{w})$. Taking the gradient with respect to $\mathbf{w}$, we have

$$\begin{aligned}
\nabla_{\mathbf{w}} f(\mathbf{w}) &= -\sum_{k \in A} \tilde{\mathcal{Q}}_k \nabla_{\mathbf{w}} \log F_k(X_k\mathbf{w}) \\
&= -\sum_{k \in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \nabla_{\mathbf{w}} F_k(X_k\mathbf{w})
\end{aligned}$$

$$= -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \left[\sum_{i=1}^{m-1} \left.\frac{\partial F_k(\mathbf{z})}{\partial z_i}\right|_{z=X_k\mathbf{w}} \cdot \nabla_{\mathbf{w}} (X_k\mathbf{w})_i \right]$$
$$= -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \left[\sum_{i=1}^{m-1} \left.\frac{\partial F_k(\mathbf{z})}{\partial z_i}\right|_{z=X_k\mathbf{w}} \cdot X_k^{(i)} \right], \tag{G.11}$$

where the third equality holds as $F_k(\mathbf{z})$ has multidimensional input and we're taking the total derivative. Hence, we need to compute $\frac{\partial F_k(\mathbf{z})}{\partial z_i}$. Writing CDF F_k in terms of its PDF p_k (which exists as $X_k X_k^\intercal$ is invertible), we have

$$F_k(\mathbf{z}) = \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_{m-1}} p_k(x_1, \ldots, x_{m-1}) dx_1 \ldots dx_{m-1}.$$

We compute partial derivative w.r.t. z_1 first, for simplicity, and generalize it after. In particular,

$$\begin{aligned}
\frac{\partial F_k(\mathbf{z})}{\partial z_1} &= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} \frac{\partial}{\partial z_1} \left[\int_{-\infty}^{z_1} p_k(x_1, \ldots, x_{m-1}) dx_1 \right] dx_2 \ldots dx_{m-1} \\
&= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_k(z_1, \ldots, x_{m-1}) dx_2 \ldots dx_{m-1} \\
&= \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_{k,-1}(x_2, \ldots, x_{m-1}|z_1) p_{k,1}(z_1) dx_2 \ldots dx_{m-1} \\
&= p_{k,1}(z_1) \int_{-\infty}^{z_2} \cdots \int_{-\infty}^{z_{m-1}} p_{k,-1}(x_2, \ldots, x_{m-1}|z_1) dx_2 \ldots dx_{m-1} \\
&= p_{k,1}(z_1) \cdot \Pr_k(Z_2 \le z_2, \ldots, Z_{m-1} \le z_{m-1} | Z_1 = z_1) \\
&= p_{k,1}(z_1) \cdot F_{k,Z_{-1}|Z_1=z_1}(\mathbf{z}_{-1}),
\end{aligned}$$

where $F_{k,Z_{-1}|Z_1=z_1}$ is the conditional CDF of the distribution F_k given the first coordinate is z_1, $p_{k,1}$ is the marginal distribution PDF of this first coordinate, and $p_{k,-1}$ is the PDF of the rest. This derivation holds for the partial derivative w.r.t. any z_i, even though it was derived for z_1. Plugging this into Equation (G.11), the gradient therefore becomes

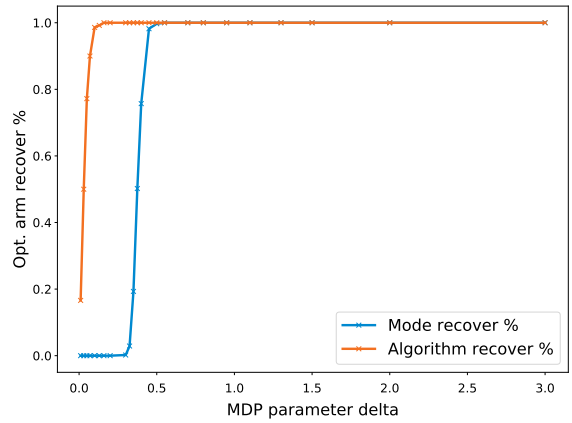
$$\nabla_{\mathbf{w}} f(\mathbf{w}) = -\sum_{k\in A} \frac{\tilde{\mathcal{Q}}_k}{F_k(X_k\mathbf{w})} \left[\sum_{i=1}^{m-1} p_{k,i}((X_k\mathbf{w})_i) \cdot F_{k,Z_{-i}|Z_i=(X_k\mathbf{w})_i}((X_k\mathbf{w})_{-i}) \cdot X_k^{(i)} \right].$$

□

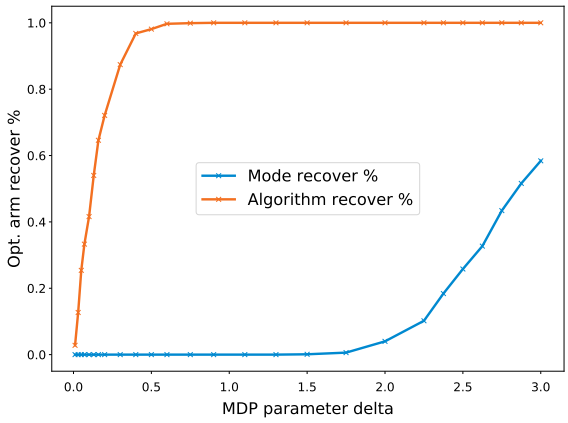
G.10 Additional Empirical Results for Inverse Bandits

G.10.1 Varying parameter δ

Here, we present the experimental results as δ is varied for additional values of σ and n. All graphs in this section have also been averaged over 1000 runs. Figure G.3 shows how

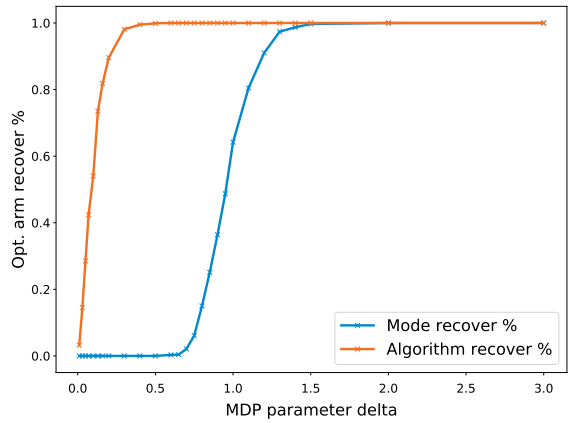


$\sigma = 0.5$

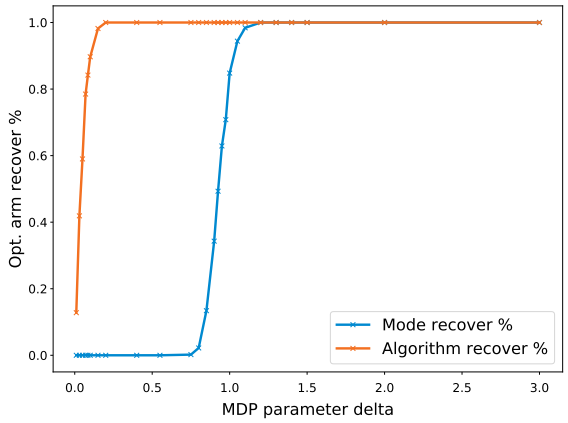


$\sigma = 2.0.$

Figure G.3: Performance as δ is varied, when σ is fixed to 0.5 and 2.



$n = 250$

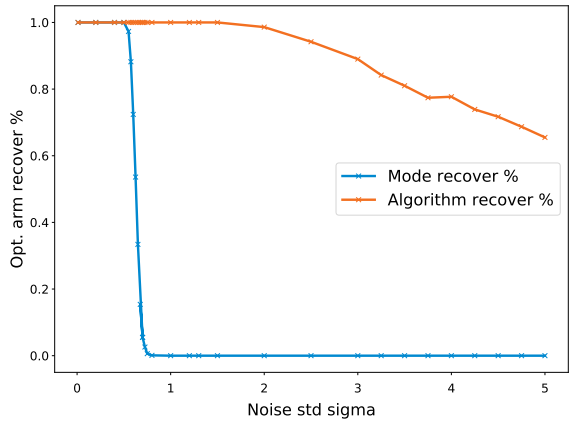


$n = 1000$

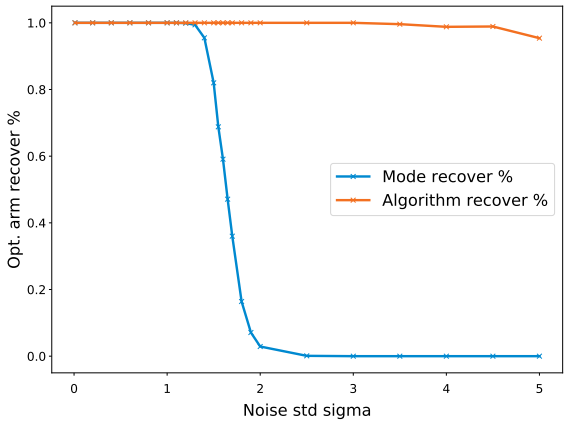
Figure G.4: Performance as δ is varied, when the number of agents is 250 and 1000.

the performance varies as δ is varied from 0.01 to 3, when σ is set to 0.5 and 2.0 (while n is still 500). As expected, one can observe that the tipping point (where the mode switches to the blue region corresponding to arm 1) occurs much earlier when $\sigma = 0.5$, and much later when $\sigma = 2$.

Figure G.4 shows how the performance varies as δ is varied from 0.01 to 3, when the number of agents n is 250 and 1000 (while σ is still set to 1). First, note that the tipping point (for the mode switch) only depends on the value of δ and σ, and indeed, we can see from the graphs that the tipping point continues to be around $\delta = 1$ irrespective of the number of the agents. But, the number of agents defines how close $\tilde{\mathcal{Q}}$ is to $\mathcal{Q}(\mathbf{w}^\star)$, and hence determines the sharpness of the transition. In particular, for a larger number of agents, the empirical mode (obtained from $\tilde{\mathcal{Q}}$) is more likely to match the true mode (of $\mathcal{Q}(\mathbf{w}^\star)$). Hence, we can see that when $n = 1000$, the transition of the mode's performance is sharper across the tipping point (because of less noise), while when $n = 250$, the transition is smoother across this tipping point (because of more noise).

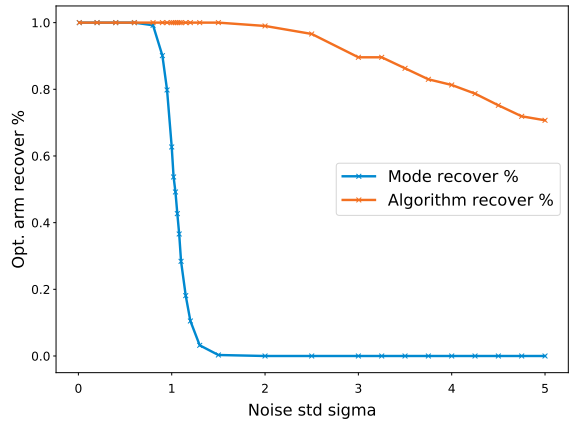


$\delta = 0.5$

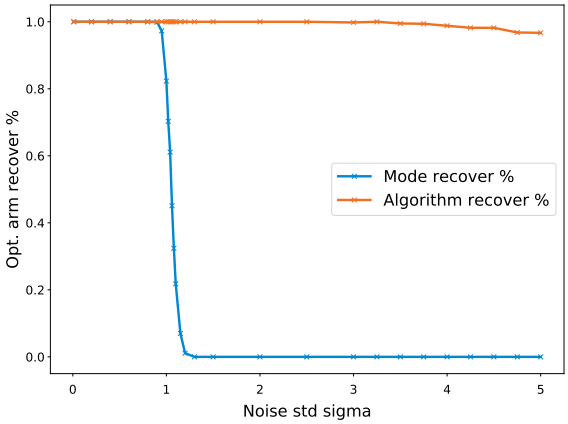


$\delta = 2.0$.

Figure G.5: Performance as σ is varied, when δ is fixed to 0.5 and 2.



$n = 250$



$n = 1000$

Figure G.6: Performance as σ is varied, when the number of agents is 250 and 1000.

G.10.2 Varying noise parameter σ

Next, we present the experimental results as σ is varied, for additional values of δ and n. All graphs in this section have also been averaged over 1000 runs. Figure G.5 shows how the performance varies as σ is varied from 0.01 to 5, when δ is set to 0.5 and 2.0 (while n is still 500). As expected, we can see that the tipping point (where the mode switches out of the blue region corresponding to arm 1) occurs earlier when $\delta = 0.5$, and much later when $\delta = 2$. Further, at high values of σ, the algorithm's performance is more robust when $\delta = 2$, as the blue region is larger.

Finally, Figure G.6 shows how the performance varies as σ is varied from 0.01 to 5, when number of agents n is 250 and 1000 (while δ is still set to 1). Again, note that the tipping point of the mode switch occurs at the same point (around $\sigma = 1$) irrespective of the number of agents. And, as Section G.10.1, when $n = 1000$, the transition of the mode's performance is sharper across the tipping point, while when $n = 250$, the transition is

smoother across it. Further, at high values of σ, $n = 1000$ has a much better algorithm performance compared to $n = 500$ (which in turn outperforms that at $n = 250$), showing that even at such high levels of noise, if $\tilde{\mathcal{Q}}$ coincides with $\mathcal{Q}(\mathbf{w}^\star)$, the algorithm is still able to recover the optimal arm 1.

Bibliography

This bibliography contains 196 references.

[AA11] M. Anderson and S. L. Anderson. *Machine Ethics*. Cambridge University Press, 2011.

[AG13] Shipra Agrawal and Navin Goyal. "Thompson Sampling for Contextual Bandits with Linear Payoffs". In: *ICML (3)*. 2013, pp. 127–135.

[Agr+18] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. "A Rewriting System for Convex Optimization Problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[AKS17] T. Arnold, D. Kasenberg, and M. Scheutzs. "Value Alignment or Misalignment - What Will Keep Systems Accountable?" In: *AI, Ethics, and Society, Papers from the 2017 AAAI Workshops*. 2017.

[Amo+16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Man. "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565* (2016).

[AN04a] P. Abbeel and A. Y. Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *Proceedings of the 21st International Conference on Machine Learning (ICML)*. 2004, pp. 1–8.

[AN04b] P. Abbeel and A. Y. Ng. "Apprenticeship learning via inverse reinforcement learning". In: *Proceedings of the 21st International Conference on Machine Learning (ICML)*. 2004.

[AO15] Christopher Amato and Frans A Oliehoek. "Scalable Planning and Learning for Multiagent POMDPs." In: *Proceedings of the AAAI Conference*. 2015.

[APX12] H. Azari Soufiani, D. C. Parkes, and L. Xia. "Random Utility Theory for Social Choice". In: *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 126–134.

[APX14a] H. Azari Soufiani, D. C. Parkes, and L. Xia. "A Statistical Decision-Theoretic Framework for Social Choice". In: *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*. Forthcoming. 2014.

[APX14b] H. Azari Soufiani, D. C. Parkes, and L. Xia. "Computing Parametric Ranking Models via Rank-Breaking". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. 2014, pp. 360–368.

[Arr51] K. Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[ASW05] C. Allen, I. Smit, and W. Wallach. "Artificial morality: Top-down, bottom-up, and hybrid approaches". In: *Ethics and Information Technology* 7.3 (2005), pp. 149–155.

[Aue+02] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. "The nonstochastic multi-armed bandit problem". In: *SIAM Journal on Computing* 32.1 (2002), pp. 48–77.

[Awa+18] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-Franois Bonnefon, and Iyad Rahwan. "The moral machine experiment". In: *Nature* 563.7729 (2018), pp. 59–64.

[Bab+11] M. Babe-Vroman, V. Marivate, K. Subramanian, and M. L. Littman. "Apprenticeship Learning About Multiple Intentions". In: *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 2011, pp. 897–904.

[Bal+12] M.-F. Balcan, F. Constantin, S. Iwata, and L. Wang. "Learning valuation functions". In: *Proceedings of the 25th Conference on Computational Learning Theory (COLT)*. 2012, pp. 4.1–4.24.

[Bal+18] A. Balakrishnan, D. Bouneffouf, N. Mattei, and F. Rossi. "Using Contextual Bandits with Behavioral Constraints for Constrained Online Movie Recommendation". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 2018.

[Bal+19a] A. Balakrishnan, D. Bouneffouf, N. Mattei, and F. Rossi. "Incorporating Behavioral Constraints in Online AI Systems". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*. 2019.

[Bal+19b] Maria-Florina Balcan, Travis Dick, Ritesh Noothigattu, and Ariel D Procaccia. "Envy-free classification". In: *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NIPS)*. 2019.

[Ber63] C. Berge. *Topological Spaces: including a treatment of multi-valued functions, vector spaces, and convexity*. Oliver & Boyd, 1963.

[BMS87] V. Bakanic, C. McPhail, and R. J. Simon. "The manuscript review and decision-making process". In: *American Sociological Review* 52.5 (1987), pp. 631–642.

[Bor+14] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. "Maximizing Social Influence in Nearly Optimal Time". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '14. Portland, Oregon: SIAM, 2014, pp. 946–957. ISBN: 978-1-611973-38-9.

[Bou+15] C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet. "Optimal Social Choice Functions: A Utilitarian View". In: *Artificial Intelligence* 227 (2015), pp. 190–213.

[Bou+17] D. Bouneffouf, I. Rish, G. A. Cecchi, and R. Fraud. "Context Attentive Bandits: Contextual Bandit with Restricted Context". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 1468–1475. DOI: 10.24963/ijcai.2017/203. URL: https://doi.org/10.24963/ijcai.2017/203.

[BR19] D. Bouneffouf and I. Rish. "A Survey on Practical Applications of Multi-Armed and Contextual Bandits". In: *CoRR* abs/1904.10040 (2019). arXiv: 1904.10040. URL: http://arxiv.org/abs/1904.10040.

[Bra+16] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, eds. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[Bra84] R. A. Bradley. "Paired Comparisons: Some Basic Procedures and Examples". In: *Handbook of Statistics*. Vol. 4. Elsevier, 1984, pp. 299–326.

[Bro+12] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. "A survey of monte carlo tree search methods". In: *IEEE Transactions on Computational Intelligence and AI in games* 4.1 (2012), pp. 1–43.

[BSR16] J.-F. Bonnefon, A. Shariff, and I. Rahwan. "The Social Dilemma of Autonomous Vehicles". In: *Science* 352.6293 (2016), pp. 1573–1576.

[BT96a] D. Bertsekas and J. Tsitsiklis. "Neuro-dynamic programming". In: *Athena Scientific* (1996).

[BT96b] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[CDC13] CDC. *HIV Surveillance Report*. www.cdc.gov/hiv/pdf/g-l/hiv_surveillance_report_vol_25.pdf. Mar. 2013.

[CGS18] S. Chatterjee, A. Guntuboyina, and B. Sen. "On matrix estimation under monotonicity constraints". In: *Bernoulli* 24.2 (2018), pp. 1072–1100.

[Cho65] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965.

[Chu05] K. Church. "Reviewing the reviewers". In: *Computational Linguistics* 31.4 (2005), pp. 575–578.

[CK12] J. Choi and K.-E. Kim. "Nonparametric Bayesian Inverse Reinforcement Learning for Multiple Reward Functions". In: *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 314–322.

[CKO01] U. Chajewska, D. Koller, and D. Ormoneit. "Learning an Agent's Utility Function by Observing Behavior". In: *Proceedings of the 18th International Conference on Machine Learning (ICML)*. 2001, pp. 35–42.

[Coh+14] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F Werneck. "Sketch-based Influence Maximization and Computation: Scaling up with guarantees". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM. 2014, pp. 629–638.

[Con11] R. Congleton. *Perfecting Parliament: Constitutional Reform, Liberalism, and the Rise of Western Democracy*. Cambridge University Press, 2011.

[Con85] Marquis de Condorcet. *Essai sur l'application de l'analyse la probabilit de dcisions rendues la pluralit de voix*. Imprimerie Royal. Facsimile published in 1972 by Chelsea Publishing Company, New York. 1785.

[CPS16] I. Caragiannis, A. D. Procaccia, and N. Shah. "When Do Noisy Votes Reveal the Truth?" In: *ACM Transactions on Economics and Computation* 4.3 (2016), article 15.

[CS05] V. Conitzer and T. Sandholm. "Common Voting Rules as Maximum Likelihood Estimators". In: *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2005, pp. 145–152.

[CS06] 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)*. 2006, pp. 627–634.

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. "When Are Elections with Few Candidates Hard to Manipulate?" In: *Journal of the ACM* 54.3 (2007), pp. 1–33.

[CT12] F. Caron and Y. W. Teh. "Bayesian Nonparametric Models for Ranked Data". In: *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 1529–1537.

[CT15] S. Chaudhuri and A. Tewari. "Online Ranking with Top-1 Feedback". In: *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2015, pp. 129–137.

[DB16] S. Diamond and S. Boyd. "CVXPY: A Python-Embedded Modeling Language for Convex Optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[Din+06] C. Ding, D. Zhou, X. He, and H. Zha. "R_1-PCA: Rotational invariant L_1-norm principal component analysis for robust subspace factorization". In: *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. 2006, pp. 281–288.

[Don+18] M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil. *Empirical Risk Minimization under Fairness Constraints*. arXiv:1802.08626. 2018.

[DSS12] A. Daniely, S. Sabato, and S. Shalev-Shwartz. "Multiclass Learning Approaches: A Theoretical Comparison with Implications". In: *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*. 2012, pp. 485–493.

[DTD15] A. Datta, M. C. Tschantz, and A. Datta. "Automated Experiments on Ad Privacy Settings: A Tale of Opacity, Choice, and Discrimination". In: *Proceedings of the 15th Privacy Enhancing Technologies Symposium (PETS)*. 2015, pp. 92–112.

[Dwo+12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. "Fairness Through Awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS)*. 2012, pp. 214–226.

[EFS09] E. Elkind, P. Faliszewski, and A. Slinko. "On Distance Rationalizability of Some Voting Rules". In: *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. 2009, pp. 108–117.

[Eve+17] T. Everitt, V. Krakovna, L. Orseau, and S. Legg. "Reinforcement Learning with a Corrupted Reward Channel". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 4705–4713.

[FEL09] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. "How Hard is Bribery in Elections?" In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 485–532.

[FEL15] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. "Weighted Electoral Control". In: *Journal of Artificial Intelligence Research* 52 (2015), pp. 507–542.

[FHS08] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. "Copeland Voting: Ties Matter". In: *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 2008, pp. 983–990.

[FMS18] B. Fain, K. Munagala, and N. Shah. "Fair Allocation of Indivisible Public Goods". In: *Proceedings of the 19th ACM Conference on Economics and Computation (EC)*. 2018, pp. 575–592.

[Fol67] D. Foley. "Resource Allocation and the Public Sector". In: *Yale Economics Essays* 7 (1967), pp. 45–98.

[Fre+20] R. Freedman, J. Schaich Borg, W. Sinnott-Armstrong, J. P. Dickerson, and V. Conitzer. "Adapting a Kidney Exchange Algorithm to Align with Human Values". In: *Artificial Intelligence* 283 (2020).

[FS95] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Proceedings of the 2nd European Conference on Computational Learning Theory (EuroCOLT)*. 1995, pp. 23–37.

[Gal+17] Y. Gal, M. Mash, A. D. Procaccia, and Y. Zick. "Which Is the Fairest (Rent Division) of Them All?" In: *Journal of the ACM* 64.6 (2017), article 39.

[GHL14] Julio Gonzlez-Daz, Ruud Hendrickx, and Edwin Lohmann. "Paired comparisons analysis: an axiomatic approach to ranking methods". In: *Social Choice and Welfare* 42.1 (2014), pp. 139–169.

[Gib73] A. Gibbard. "Manipulation of Voting Schemes". In: *Econometrica* 41 (1973), pp. 587–602.

[Gib77] A. Gibbard. "Manipulation of schemes that mix voting with chance". In: *Econometrica* 45 (1977), pp. 665–681.

[GK11] Daniel Golovin and Andreas Krause. "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization". In: *Journal of Artificial Intelligence Research* 42 (2011), pp. 427–486.

[GKP02] Carlos Guestrin, Daphne Koller, and Ronald Parr. "Multiagent planning with factored MDPs". In: *Advances in neural information processing systems*. 2002, pp. 1523–1530.

[GLD00] R. Givan, S. Leach, and T. Dean. "Bounded-Parameter Markov Decision Processes". In: *Artificial Intelligence* 122.1–2 (2000), pp. 71–109.

[GS09] J. Guiver and E. Snelson. "Bayesian Inference for Plackett-Luce Ranking Models". In: *Proceedings of the 26th International Conference on Machine Learning (ICML)*. 2009, pp. 377–384.

[GW07] F. Gao and J. A. Wellner. "Entropy estimate for high-dimensional monotonic functions". In: *Journal of Multivariate Analysis* 98.9 (2007), pp. 1751–1764.

[GWG12] Sharad Goel, Duncan J Watts, and Daniel G Goldstein. "The structure of online diffusion networks". In: *Proceedings of the 13th ACM conference on electronic commerce*. ACM. 2012, pp. 623–638.

[Had+16] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. D. Dragan. "Cooperative Inverse Reinforcement Learning". In: *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 3909–3917.

[Hb+18] . Hbert-Johnson, M. P. Kim, O. Reingold, and G. N. Rothblum. "Calibration for the (Computationally-Identifiable) Masses". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. Forthcoming. 2018.

[HGC03] M. Hojat, J. S. Gonnella, and A. S. Caelleigh. "Impartial judgment by the "gatekeepers" of science: Fallibility and accountability in the peer review process". In: *Advances in Health Sciences Education* 8.1 (2003), pp. 75–96.

[HL50] J. L. Hodges Jr and E. L. Lehmann. "Some problems in minimax point estimation". In: *The Annals of Mathematical Statistics* (1950), pp. 182–197.

[HNP18] Nika Haghtalab, Ritesh Noothigattu, and Ariel D Procaccia. "Weighted voting via no-regret learning". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[HPS16] M. Hardt, E. Price, and N. Srebro. "Equality of Opportunity in Supervised Learning". In: *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 3315–3323.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gal Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11–15.

[Jar85] J. Jarvis Thomson. "The Trolley Problem". In: *The Yale Law Journal* 94.6 (1985), pp. 1395–1415.

[Jos+16] M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. "Fairness in Learning: Classic and Contextual Bandits". In: *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 325–333.

[JR11] G. A. Jehle and P. J. Reny. *Advanced Microeconomic Theory*. Pearson, Prentice Hall, 2011.

[Kah+19] A. Kahng, M. K. Lee, R. Noothigattu, A. D. Procaccia, and C.-A. Psomas. "Statistical Foundations of Virtual Democracy". In: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2019, pp. 3173–3182.

[KBP13] J. Kober, J. A. Bagnell, and J. Peters. "Reinforcement Learning in Robotics: A Survey". In: *International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274.

[KDH11] D. Kong, C. Ding, and H. Huang. "Robust nonnegative matrix factorization using L21-norm". In: *Proceedings of the 20th International Conference on Information and Knowledge Management (CIKM)*. 2011, pp. 673–682.

[Kea+18] M. Kearns, S. Neel, A. Roth, and S. Wu. "Computing Parametric Ranking Models via Rank-Breaking". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 2018.

[KHL08] Hanna Kurniawati, David Hsu, and Wee Sun Lee. "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces." In: *Robotics: Science and systems*. Vol. 2008. Zurich, Switzerland. 2008.

[Kil+17] N. Kilbertus, M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schlkopf. "Avoiding Discrimination through Causal Reasoning". In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 656–666.

[KKT03] David Kempe, Jon Kleinberg, and va Tardos. "Maximizing the Spread of Influence through a Social Network". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2003, pp. 137–146.

[KLM96] L. P. Kaelbling, M. L. Littman, and A. W. Moore. "Reinforcement Learning: A Survey". In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285.

[KS06] Levente Kocsis and Csaba Szepesvri. "Bandit based monte-carlo planning". In: *European conference on machine learning*. Springer. 2006, pp. 282–293.

[KTP77] S. Kerr, J. Tolliver, and D. Petree. "Manuscript characteristics which influence acceptance for management and social science journals". In: *Academy of Management Journal* 20.1 (1977), pp. 132–141.

[Lam09] M. Lamont. *How Professors Think*. Harvard University Press, 2009.

[Lee+19] M. K. Lee et al. "WeBuildAI: Participatory Framework for Fair and Efficient Algorithmic Governance". In: *Proceedings of the 22nd ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*. article 181. 2019.

[Lee15] E. Lee. *APX-Hardness of Maximizing Nash Social Welfare with Indivisible Items*. arXiv:1507.01159. 2015.

[Lei+17] J. Leike, M. Martic, V. Krakovna, P.A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg. "AI safety gridworlds". In: *arXiv preprint arXiv:1711.09883* (2017).

[LF17] Romain Laroche and Raphael Feraud. "Reinforcement Learning Algorithm Selection". In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. 2017.

[Liu+18] Guiliang Liu, Oliver Schulte, Wang Zhu, and Qingcan Li. "Toward Interpretable Deep Reinforcement Learning with Linear Model U-Trees". In: *CoRR* abs/1807.05887 (2018). arXiv: 1807.05887. URL: http://arxiv.org/abs/1807.05887.

[Lor+18a] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. "Preferences and Ethical Principles in Decision Making". In: *Proc. of the 1st AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2018.

[Lor+18b] A. Loreggia, N. Mattei, F. Rossi, and K. B. Venable. "Value Alignment Via Tractable Preference Distance". In: *Artificial Intelligence Safety and Security*. Ed. by R. V. Yampolskiy. CRC Press, 2018. Chap. 18.

[LP16] Ronny Luss and Marek Petrik. "Interpretable Policies for Dynamic Product Recommendations". In: *Proc. Conf. Uncertainty Artif. Intell.* New York, USA, June 2016, p. 74.

[LRT11] B. T. Luong, S. Ruggieri, and F. Turini. "k-NN as an Implementation of Situation Testing for Discrimination Discovery and Prevention". In: *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2011, pp. 502–510.

[Luc59] R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

[LZ08] John Langford and Tong Zhang. "The Epoch-Greedy Algorithm for Contextual Multi-armed Bandits". In: *Proc. 21st NIPS*. 2008.

[MA08] J. C. Mogul and T. Anderson. "Before and After WOWCS: A literature survey, A list of papers we wish had been submitted". In: *WOWCS*. 2008. URL: %7Bhttps://www.usenix.org/legacy/events/wowcs/tech/full_papers/b4after/b4after_html/index.html%7D.

[Mah77] M. J. Mahoney. "Publication prejudices: An experimental study of confirmatory bias in the peer review system". In: *Cognitive Therapy and Research* 1.2 (1977), pp. 161–175.

[Mar95] J. I. Marden. *Analysing and Modeling Rank Data*. Chapman & Hall, 1995.

[MBM18] S. Mei, Y. Bai, and A. Montanari. "The landscape of empirical risk for non-convex losses". In: *Annals of Statistics* 46.6A (2018), pp. 2747–2774.

[MG15] L. Maystre and M. Grossglauser. "Fast and Accurate Inference of Plackett-Luce Models". In: *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS)*. 2015, pp. 172–180.

[Mik11] J. Mikhail. *Elements of Moral Cognition: Rawls' Linguistic Analogy and the Cognitive Science of Moral and Legal Judgment*. Cambridge University Press, 2011.

[Mil+09] Norweeta G Milburn, Eric Rice, Mary Jane Rotheram-Borus, Shelley Mallett, Doreen Rosenthal, Phillip Batterham, Susanne J May, Andrea Witkin, and Naihua Duan. "Adolescents Exiting Homelessness over two years: The Risk Amplification and Abatement Model". In: *Journal of Research on Adolescence* 19.4 (2009), pp. 762–785.

[Mni+15] V. Mnih et al. "Human-Level Control Through Deep Reinforcement Learning". In: *Nature* 518 (2015), pp. 529–533.

[Mos51] F. Mosteller. "Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations". In: *Psychometrika* 16.1 (1951), pp. 3–9.

[Mou03] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.

[Mou83] H. Moulin. *The Strategy of Social Choice*. Vol. 18. Advanced Textbooks in Economics. North-Holland, 1983.

[MS17] P. Manurangsi and W. Suksompong. "Asymptotic Existence of Fair Divisions for Groups". In: *Mathematical Social Sciences* 89 (2017), pp. 100–108.

[MV08] H. Masnadi-Shirazi and N. Vasconcelos. "On the design of loss functions for classification: Theory, robustness to outliers, and SavageBoost". In: *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*. 2008, pp. 1049–1056.

[Mye95] R. B. Myerson. "Axiomatic derivation of scoring rules without the ordering assumption". In: *Social Choice and Welfare* 12.1 (1995), pp. 59–74.

[Nat89] B. K. Natarajan. "On learning sets and functions". In: *Machine Learning* 4.1 (1989), pp. 67–97.

[Nie+10] F. Nie, H. Huang, X. Cai, and C. H. Ding. "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization". In: *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS)*. 2010, pp. 1813–1821.

[NJ04] T. D. Nielsen and F. V. Jensen. "Learning a Decision Maker's Utility Function From (Possibly) Inconsistent Behavior". In: *Artificial Intelligence* 160.1–2 (2004), pp. 53–78.

[Noo+18] R. Noothigattu, S. S. Gaikwad, E. Awad, S. Dsouza, I. Rahwan, P. Ravikumar, and A. D. Procaccia. "A Voting-Based System for Ethical Decision Making". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 1587–1594.

[Noo+19] R. Noothigattu, D. Bouneffouf, N. Mattei, R. Chandra, P. Madan, K. Varshney, M. Campbell, M. Singh, and F. Rossi. "Teaching AI Agents Ethical Values Using Reinforcement Learning and Policy Orchestration". In: *IBM Journal of Research & Development* (2019).

[NPP20] Ritesh Noothigattu, Dominik Peters, and Ariel D Procaccia. *Axioms for Learning from Pairwise Comparisons*. Manuscript. 2020.

[NR00a] A. Y. Ng and S. Russell. "Algorithms for Inverse Reinforcement Learning". In: *Proceedings of the 17th International Conference on Machine Learning (ICML)*. 2000, pp. 663–670.

[NR00b] Andrew Y. Ng and Stuart J. Russell. "Algorithms for Inverse Reinforcement Learning". In: *Proceedings of the Seventeenth International Conference on Machine Learning*. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670. ISBN: 1-55860-707-2. URL: http://dl.acm.org/citation.cfm?id=645529.657801.

[NSP20] Ritesh Noothigattu, Nihar B Shah, and Ariel D Procaccia. "Loss Functions, Axioms, and Peer Review". In: *ICML Workshop on Incentives in Machine Learning*. 2020.

[NYP20] Ritesh Noothigattu, Tom Yan, and Ariel D Procaccia. *Inverse Reinforcement Learning From Like-Minded Teachers*. Manuscript. 2020.

[Pla75] R. Plackett. "The analysis of permutations". In: *Applied Statistics* 24 (1975), pp. 193–202.

[PM02] F. Perron and E. Marchand. "On the minimax estimator of a bounded normal mean". In: *Statistics and Probability Letters* 58 (2002), pp. 327–333.

[PPR15] A. Prasad, H. H. Pareek, and P. Ravikumar. "Distributional Rank Aggregation, and an Axiomatic Analysis". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2015, pp. 2104–2112.

[Pro10] A. D. Procaccia. "Can Approximation Circumvent Gibbard-Satterthwaite?" In: *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*. 2010, pp. 836–841.

[Pro13] A. D. Procaccia. "Cake Cutting: Not Just Child's Play". In: *Communications of the ACM* 56.7 (2013), pp. 78–87.

[PSZ16] A. D. Procaccia, N. Shah, and Y. Zick. "Voting rules as error-correcting codes". In: *Artificial Intelligence* 231 (2016), pp. 1–16.

[Put09] Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2009.

[RA14] A. Rajkumar and S. Agarwal. "A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. 2014, pp. 118–126.

[Raw71] J. Rawls. *A Theory of Justice*. Harvard University Press, 1971.

[RB09] K. Regan and C. Boutilier. "Regret-based Reward Elicitation for Markov Decision Processes". In: *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2009, pp. 444–451.

[RB10] K. Regan and C. Boutilier. "Robust Policy Computation in Reward-uncertain MDPs using Nondominated Policies". In: *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*. 2010, pp. 1127–1133.

[RDT15] S. Russell, D. Dewey, and M. Tegmark. "Research Priorities for Robust and Beneficial Artificial Intelligence". In: *AI Magazine* 36.4 (2015), pp. 105–114.

[Ric+12a] Eric Rice, Anthony Fulginiti, Hailey Winetrobe, Jorge Montoya, Aaron Plant, and Timothy Kordic. "Sexuality and Homelessness in Los Angeles public schools". In: *American Journal of Public Health* 102 (2012).

[Ric+12b] Eric Rice, Eve Tulbert, Julie Cederbaum, Anamika Barman Adhikari, and Norweeta G Milburn. "Mobilizing Homeless Youth for HIV Prevention: a Social Network Analysis of the Acceptability of a face-to-face and Online Social Networking Intervention". In: *Health education research* 27.2 (2012), p. 226.

[Ric10] Eric Rice. "The Positive Role of Social Networks and Social Networking Technology in the Condom-using Behaviors of Homeless Young People". In: *Public health reports* 125.4 (2010), p. 588.

[RKJ08] F. Radlinski, R. Kleinberg, and T. Joachims. "Learning diverse rankings with multi-armed bandits". In: *Proceedings of the 25th International Conference on Machine Learning (ICML)*. 2008, pp. 784–791.

[RM19] F. Rossi and N. Mattei. "Building Ethically Bounded AI". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*. 2019.

[Ros+04] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri. "Are loss functions all the same?" In: *Neural Computation* 16.5 (2004), pp. 1063–1076.

[RR13] Eric Rice and Harmony Rhoades. "How Should Network-based Prevention for Homeless Youth be Implemented?" In: *Addiction* 108.9 (2013), p. 1625.

[RW98] J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998.

[RY18] G. N. Rothblum and G. Yona. *Probably Approximately Metric-Fair Learning*. arXiv:1803.03242. 2018.

[Sat75] M. Satterthwaite. "Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions". In: *Journal of Economic Theory* 10 (1975), pp. 187–217.

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[SB17] Richard S. Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. 2nd. MIT Press, 2017.

[SB98a] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[SB98b] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.

[SBW18] N. B. Shah, S. Balakrishnan, and M. J. Wainwright. *Low Permutation-rank Matrices: Structural Properties and Noisy Completion*. arXiv:1709.00127. 2018.

[Sen71] A. K. Sen. "Choice Functions and Revealed Preference". In: *Review of Economic Studies* 38.3 (1971), pp. 307–317.

[Sen74] A. Sen. "Choice, Ordering and Morality". In: *Practical Reason*. Ed. by S. Krner. Oxford: Blackwell, 1974.

[Sha+16] N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. "Stochastically Transitive Models for Pairwise Comparisons: Statistical and Computational Issues". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. 2016, pp. 11–20.

[Sil+16] D. Silver et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529 (2016), pp. 484–489.

[Sim18] T. Simonite. "When Bots Teach Themselves to Cheat". In: *Wired Magazine* (Aug. 2018).

[Sin12] Yaron Singer. "How to win friends and influence people, truthfully: influence maximization mechanisms for social networks". In: *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM. 2012, pp. 733–742.

[SKP12] C Seshadhri, Tamara G Kolda, and Ali Pinar. "Community Structure and Scale-free Collections of Erds-Rnyi Graphs". In: *Physical Review E* 85.5 (2012), p. 056109.

[Smi73] J. H. Smith. "Aggregation of Preferences with Variable Electorate". In: *Econometrica* 41.6 (1973), pp. 1027–1041.

[SOA17] Katt Sammie, Frans Oliehoek, and Christopher Amato. "Learning in POMDPs with Monte Carlo Tree Search". In: *ICML17*. Aug. 2017, pp. 1819–1827.

[Spa96] P. G. Spain. "The Fermat point of a triangle". In: *Mathematics Magazine* 69.2 (1996), pp. 131–133.

[SS08] U. Syed and R. E. Schapire. "A game-theoretic approach to apprenticeship learning". In: *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*. 2008, pp. 1449–1456.

[Su99] F. E. Su. "Rental Harmony: Sperner's Lemma in Fair Division". In: *American Mathematical Monthly* 106.10 (1999), pp. 930–942.

[SV10] David Silver and Joel Veness. "Monte-Carlo Planning in large POMDPs". In: *Advances in Neural Information Processing Systems*. 2010, pp. 2164–2172.

[Swe13] L. Sweeney. "Discrimination in Online Ad Delivery". In: *Communications of the ACM* 56.5 (2013), pp. 44–54.

[SZL15] John A Schneider, A Ning Zhou, and Edward O Laumann. "A new HIV Prevention Network Approach: Sociometric Peer Change Agent Selection". In: *Social Science & Medicine* 125 (2015), pp. 192–202.

[Thu27] L. L. Thurstone. "A law of comparative judgement". In: *Psychological Review* 34 (1927), pp. 273–286.

[TWB16] Andreas Theodorou, Robert H Wortham, and Joanna J Bryson. "Why is my robot behaving like that? Designing transparency for real time inspection of autonomous robots". In: *AISB Workshop on Principles of Robotics*. University of Bath. 2016.

[TXS14] Youze Tang, Xiaokui Xiao, and Yanchen Shi. "Influence maximization: Near-Optimal Time Complexity meets Practical Efficiency". In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM. 2014, pp. 75–86.

[Var74] H. Varian. "Equity, envy and efficiency". In: *Journal of Economic Theory* 9 (1974), pp. 63–91.

[VC71] V. Vapnik and A. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities". In: *Theory of Probability and its Applications* 16.2 (1971), pp. 264–280.

[Ver+18] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. "Programmatically Interpretable Reinforcement Learning". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*. 2018, pp. 5052–5061. URL: http://proceedings.mlr.press/v80/verma18a.html.

[VG18] D. Ventura and D. Gates. "Ethics as Aesthetic: A Computational Creativity Approach to Ethical Behavior". In: *Proc. Int. Conference on Computational Creativity (ICCC)*. 2018, pp. 185–191.

[VP07] Thomas W Valente and Patchareeya Pumpuang. "Identifying Opinion Leaders to Promote Behavior Change". In: *Health Education & Behavior* (2007).

[WA08] W. Wallach and C. Allen. *Moral Machines: Teaching Robots Right From Wrong*. Oxford University Press, 2008.

[Wai19] M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press, 2019.

[Wil+17] Bryan Wilder, Amulya Yadav, Nicole Immorlica, Eric Rice, and Milind Tambe. "Uncharted but not Uninfluenced: Influence Maximization with an uncertain network". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2017, pp. 1305–1313.

[WL18] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 1687–1694.

[Woo+17] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro. "Learning Non-Discriminatory Predictors". In: *Proceedings of the 30th Conference on Computational Learning Theory (COLT)*. 2017, pp. 1920–1953.

[Xia16] L. Xia. "Bayesian Estimators As Voting Rules". In: *Proceedings of the 32nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 2016.

[Yad+16] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. "Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty". In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2016, pp. 740–748.

[Yad+17] Amulya Yadav, Bryan Wilder, Eric Rice, Robin Petering, Jaih Craddock, Amanda Yoshioka-Maxwell, Mary Hemler, Laura Onasch-Vera, Milind Tambe, and Darlene Woo. "Influence maximization in the field: The arduous journey from emerging to deployed application". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2017, pp. 150–158.

[Yad+18] Amulya Yadav, Ritesh Noothigattu, Eric Rice, Laura Onasch-Vera, Leandro Soriano Marcolino, and Milind Tambe. "Please be an Influencer? Contingency-Aware Influence Maximization". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2018, pp. 1423–1431.

[Yao77] A. C. Yao. "Probabilistic Computations: Towards a Unified Measure of Complexity". In: *Proceedings of the 17th Symposium on Foundations of Computer Science (FOCS)*. 1977, pp. 222–227.

[You75] H. P. Young. "Social Choice Scoring Functions". In: *SIAM Journal of Applied Mathematics* 28.4 (1975), pp. 824–838.

[Yu+18] H. Yu, Z. Shen, C. Miao, C. Leung, V. R. Lesser, and Q. Yang. "Building Ethics into Artificial Intelligence." In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. 2018, pp. 5527–5533.

[Zaf+17] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, K. P. Gummadi, and A. Weller. "From Parity to Preference-based Notions of Fairness in Classification". In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 228–238.

[Zem+13] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. "Learning Fair Representations". In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*. 2013, pp. 325–333.

[Zie+08] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. "Maximum Entropy Inverse Reinforcement Learning". In: *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*. 2008, pp. 1433–1438.

[Zie10] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. thesis, Carnegie Mellon University. 2010.

[ZLN14] J. Zheng, S. Liu, and L. M. Ni. "Robust Bayesian Inverse Reinforcement Learning with Sparse Behavior Noise". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*. 2014, pp. 2198–2205.

[ZPR09] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. "Algorithms for the Coalitional Manipulation Problem". In: *Artificial Intelligence* 173.2 (2009), pp. 392–412.